

Group-Structured and Independent Subspace Based Dictionary Learning

Zoltán Szabó

Eötvös Loránd University

Supervisor: András Lőrincz
Senior Researcher, CSc

Ph.D. School of Mathematics
Miklós Laczkovich

Applied Mathematics Programme
György Michaletzky

Budapest, 2011.

Contents

1	Introduction	1
2	Theory – Group-Structured Dictionary Learning	11
2.1	Online Group-Structured Dictionary Learning	11
2.1.1	Problem Definition	11
2.1.2	Optimization	16
2.2	Generalized Support Vector Machines and ϵ -Sparse Representations	20
2.2.1	Reproducing Kernel Hilbert Space	20
2.2.2	Support Vector Machine	24
2.2.3	Equivalence of Generalized Support Vector Machines and ϵ -Sparse Coding	25
2.3	Multilayer Kerceptron	28
2.3.1	Multilayer Perceptron	29
2.3.2	The Multilayer Kerceptron Architecture	29
2.3.3	Backpropagation of Multilayer Kerceptrons	31
3	Theory – Independent Subspace Based Dictionary Learning	35
3.1	Controlled Models	36
3.1.1	D-optimal Identification of ARX Models	36
3.1.2	The ARX-IPA Problem	37
3.1.3	Identification Method for ARX-IPA	38
3.2	Incompletely Observable Models	39
3.2.1	The AR-IPA Model with Missing Observations	40
3.2.2	Identification Method for mAR-IPA	40
3.3	Complex Models	41

3.3.1	Complex Random Variables	41
3.3.2	Complex Independent Subspace Analysis	42
3.3.3	Identification Method for Complex ISA	43
3.4	Nonparametric Models	43
3.4.1	Functional Autoregressive Independent Process Analysis	44
3.4.2	Identification Method for fAR-IPA	45
3.5	Convolutive Models	46
3.5.1	Complete Blind Subspace Deconvolution	47
3.5.2	Identification Method for Complete BSSD	48
3.6	Information Theoretical Estimations via Random Projections	48
4	Numerical Experiments – Group-Structured Dictionary Learning	51
4.1	Inpainting of Natural Images	51
4.2	Online Structured Non-negative Matrix Factorization on Faces	56
4.3	Collaborative Filtering	58
4.3.1	Collaborative Filtering as Structured Dictionary Learning	59
4.3.2	Neighbor Based Correction	59
4.3.3	Numerical Results	60
5	Numerical Experiments – Independent Subspace Based Dictionary Learning	71
5.1	Test Datasets	71
5.2	Performance Measure, the Amari-index	73
5.3	Numerical Results	75
5.3.1	ARX-IPA Experiments	77
5.3.2	mAR-IPA Experiments	78
5.3.3	Complex ISA Experiments	81
5.3.4	fAR-IPA Experiments	84
5.3.5	Complete BSSD Experiments	89
5.3.6	ISA via Random Projections	90
6	Conclusions	97

A	Proofs	99
A.1	Online Group-Structured Dictionary Learning	99
A.1.1	The Forgetting Factor in Matrix Recursions	99
A.1.2	Online Update Equations for the Minimum Point of \hat{f}_t . . .	100
A.2	Correspondence of the (c, e)-SVM and (p, s)-Sparse Problems . . .	104
A.3	Backpropagation for Multilayer Kerceptrons	107
A.3.1	Derivative of the Approximation Term	107
A.3.2	Derivative of the Regularization Term	112
A.3.3	Derivative of the Cost	113
B	Abbreviations	114
	Short Summary in English	138
	Short Summary in Hungarian	139

Acknowledgements

I would like to express my thanks to my supervisor András Lőrincz for his inspiring personality and the continuous support. I owe the members of his research group thank for the friendly atmosphere. I'm especially grateful to Barnabás Póczos for our discussions. I'm deeply indebted to my family for the peaceful background. Thanks to my niece and nephew for continuously stimulating my mind with their joy.

Abstract

Thanks to the several successful applications, sparse signal representation has become one of the most actively studied research areas in mathematics. However, in the traditional sparse coding problem the dictionary used for representation is assumed to be known. In spite of the popularity of sparsity and its recently emerging structured sparse extension, interestingly, very few works focused on the learning problem of dictionaries to these codes.

In the first part of the paper, we develop a dictionary learning method which is (i) online, (ii) enables overlapping group structures with (iii) non-convex sparsity-inducing regularization and (iv) handles the partially observable case. To the best of our knowledge, current methods can exhibit two of these four desirable properties at most. We also investigate several interesting special cases of our framework and demonstrate its applicability in inpainting of natural signals, structured sparse non-negative matrix factorization of faces and collaborative filtering. Complementing the sparse direction we formulate a novel component-wise acting, ϵ -sparse coding scheme in reproducing kernel Hilbert spaces and show its equivalence to a generalized class of support vector machines. Moreover, we embed support vector machines to multilayer perceptrons and show that for this novel kernel based approximation approach the backpropagation procedure of multilayer perceptrons can be generalized.

In the second part of the paper, we focus on dictionary learning making use of *independent subspace* assumption instead of *structured sparsity*. The corresponding problem is called independent subspace analysis (ISA), or independent component analysis (ICA) if all the hidden, independent sources are one-dimensional. One of the most fundamental results of this research field is the ISA separation principle, which states that the ISA problem can be solved by traditional ICA up to permutation. This principle (i) forms the basis of the state-of-the-art ISA solvers and (ii) enables one to estimate the unknown number and the dimensions of the sources efficiently. We (i) extend the ISA problem to several new directions including the controlled, the partially observed, the complex valued and the nonparametric case

and (ii) derive separation principle based solution techniques for the generalizations. This solution approach (i) makes it possible to apply state-of-the-art algorithms for the obtained subproblems (in the ISA example ICA and clustering) and (ii) handles the case of unknown dimensional sources. Our extensive numerical experiments demonstrate the robustness and efficiency of our approach.

Chapter 1

Introduction

Sparse signal representation is among the most actively studied research areas in mathematics. In the *sparse coding* framework one approximates the observations with the linear combination of a few vectors (basis elements) from a *fixed dictionary* [1, 2]. The general sparse coding problem, i.e., the ℓ_0 -norm solution that searches for the least number of basis elements, is NP-hard [3]. To overcome this difficulty, a popular approach is to apply ℓ_p ($0 < p \leq 1$) relaxations. The $p = 1$ special case, the so-called Lasso problem [4], has become particularly popular since in this case the relaxation leads to a convex problem.

The traditional form of sparse coding does not take into account any prior information about the structure of hidden representation (also called covariates, or code). However, using *structured sparsity* [5–60], that is, forcing different kind of structures (e.g., disjunct groups, trees, or more general overlapping group structures) on the sparse codes can lead to increased performances in several applications. Indeed, as it has been theoretically proved recently structured sparsity can ease feature selection [5, 6], and makes possible robust compressed sensing with substantially decreased observation number [7]. Many other real life applications also confirm the benefits of structure sparsity, for example (i) automatic image annotation [8], (ii) group-structured feature selection for micro array data processing [9–19, 43], (iii) multi-task learning problems (a.k.a. transfer learning, joint covariate/subspace selection, multiple measurements vector model, simultaneous sparse approximation) [20–27, 47], (iv) fMRI (functional magnetic resonance imag-

ing) analysis [42], (v) multiple kernel learning [22, 44–46, 48, 61], (vi) analysis of soil diversity and forest diversity associations [28], (vii) handwriting, satellite and natural image classification [29–31], (viii) facial expression discrimination [32] and face recognition [33], (ix) graph labelling [34], (x) compressive imaging [35–38], (xi) structure learning in graphical models [39, 40], (xi) multi-view learning (human pose estimation) [41], (xii) natural language processing [30].

All the above mentioned examples only consider the structured sparse coding problem, where we assume that the dictionary is already given and available to us. A more interesting (and challenging) problem is the combination of these two tasks, i.e., learning the best structured dictionary and structured representation. This is the *structured dictionary learning* (SDL) problem, for which one can find only a few solutions in the literature [62–67]. Tree based group structure is assumed in [62], and dictionary learning is accomplished by means of the so-called proximal methods [68]. The efficiency of *non-convex sparsity-inducing* norms on the dictionary has recently been demonstrated in structured sparse PCA (principal component analysis) [63] in case of *general group structures*. General group-structured, but convex sparsity-inducing regularizer is applied in [64] for the learning of the dictionary by taking advantage of network flow algorithms. In [65], the authors take partition (special group structure) on the hidden covariates and explicitly limit the number of non-zero elements in each group in the dictionary learning problem. Dictionary learning is carried out under the assumption of one-block sparsity for the representation (special partition group structure with one active partition element) in [67], however in contrast to the previous works the approach is blind, that is it can handle *missing observations*. The cost function based on structure-inducing regularization in [66] is a special case of [63]. However, as opposed to the previous works, in [66] the presented dictionary learning approach is *online*, allowing a continuous flow of observations.

This novel SDL field is appealing for (i) transformation invariant feature extraction [66], (ii) image denoising/inpainting [62, 64, 67], (iii) background subtraction [64], (iv) analysis of text corpora [62], and (v) face recognition [63].

We are interested in structured dictionary learning algorithms that possess the following four properties:

- They can handle general, overlapping group structures.

- The applied regularization can be non-convex and hence allow less restrictive assumptions on the problem. Indeed, as it has been recently shown in the sparse coding literature:
 - by replacing the ℓ_1 norm with the ℓ_p ($0 < p < 1$) non-convex quasi-norm, exact reconstruction of the sparse codes is possible with substantially fewer measurements [69, 70].
 - The ℓ_p based approach (i) provides recovery under weaker RIP (restrictive isometry property) conditions on the dictionary than the ℓ_1 technique, (ii) moreover it inherits the robust recovery property of the ℓ_1 method with respect to the noise and the compressibility of the code [71, 72].
 - Similar properties also hold for certain more general non-convex penalties [73–76].
- We want online algorithms [66, 77, 78]:
 - Online methods have the advantage over offline ones that they can process more instances in the same amount of time [79], and in many cases this can lead to increased performance.
 - In large systems where the whole dataset does not fit into the memory, online systems can be the only solutions.
 - Online techniques are adaptive: for example in recommender systems [80] when new users appear, we might not want to relearn the dictionary from scratch; we simply want to modify it by the contributions of the new users.
- We want an algorithm that can handle missing observations [67, 81]. Using a collaborative filtering [80] example, users usually do not rate every item, and thus some of the possible observations are missing.

Unfortunately, existing approaches in the literature can possess only two of our four requirements at most. Our **first goal** (Section 2.1) is to formulate a general structured dictionary learning approach, which is (i) online, (ii) enables overlapping group structures with (iii) non-convex group-structure inducing regularization,

and (iv) handles the partially observable case. We call this problem *online group-structured dictionary learning* (OSDL).

Traditional sparse coding schemes work in the finite dimensional Euclidean space. Interestingly, however the sparse coding approach can also be extended to a more general domain, to reproducing kernel Hilbert spaces (RKHS) [82]. Moreover, as it has been proved recently [83, 84] certain variants of the sparse coding problems in RKHSs are equivalent to one of the most successful, kernel based approximation technique, the support vector machine (SVM) approach [85, 86]. Application of kernels:

- makes it possible to generalize a wide variety of linear problems to the non-linear domain thanks to the scalar product evaluation property of kernels, the so-called ‘kernel trick’.
- provides a uniform framework for numerous well-known approximation schemes, e.g., Fourier, polynomial, wavelet approximations.
- allows to define similarity measures for structured objects like strings, genes, graphs or dynamical systems.

For a recent review on kernels and SVMs, see [87]. In that cited works [83, 84], however the ϵ -insensitivity parameter of the SVMs—which only penalizes the deviations from the target value larger than ϵ , linearly—was transformed into ‘uniform’ sparsification, in the sense that ϵ was transformed to the weight of the sparsity-inducing regularization term. Our question was, whether it is possible to transform the insensitivity ϵ into a component-wise acting, ϵ -sparse scheme. Our **second goal** was to answer this kernel based sparse coding problem. We focus on this topic and give positive answer to this novel sparse coding – kernel based function approximation equivalence in Section 2.2.

Beyond SVMs, multilayer perceptron (MLP) are among the most well-known and successful approximation techniques. The basic idea of the MLP neural network is to approximate the target function, which is given to us in the form of input-output pairs, as a composition of ‘simple’ functions. In the traditional form of MLPs one assumes at each layer of the network (that is for the functions constituting the composition) a linear function followed by a component-wise acting sigmoid

function. The parameter tuning of MLP can be carried out by the backpropagation technique. For an excellent review on neural networks and MLPs, see [88]. However, MLPs consider transformations only in the finite dimensional Euclidean space at each hidden layer. Our **third goal** was to extend the scope of MLPs to the more general RKHS construction. This novel kernel based approximation scheme, the multilayer kerceptron network and the derivation of generalized backpropagation rules will be in the focus of Section 2.3.

Till now (Chapter 2) we focused on different structured sparse dictionary learning problems, and the closely related sparse coding, kernel approximation schemes. However, the dictionary learning task, (a.k.a. matrix factorization [89]) is a general problem class that contains, e.g., (sparse) principal component analysis (PCA) [90], independent component analysis (ICA) [91], independent subspace analysis (ISA) [92]¹, and (sparse) non-negative matrix factorization (NMF) [94–96], among many others. In the second part the paper (Chapter 3) we are dealing with *independent subspace* based dictionary learning, i.e., extensions of independent subspace analysis.

One predecessor of ISA is the ICA task. Independent component analysis [97, 98] has received considerable attention in signal processing and pattern recognition, e.g., in face representation and recognition [99, 100], information theoretical image matching [101], feature extraction of natural images [102], texture segmentation [103], artifact separation in MEG (magneto-encephalography) recordings and the exploration of hidden factors in financial data [104]. One may consider ICA as a cocktail party problem: we have some speakers (sources) and some microphones (sensors), which measure the mixed signals emitted by the sources. The task is to estimate the original sources from the mixed observations only. For a recent review about ICA, see [91, 105, 106].

Traditional ICA algorithms are *one-dimensional* in the sense that all sources are assumed to be independent *real* valued random variables. Nonetheless, applications in which only certain groups of the hidden sources are independent may be highly relevant in practice, because one cannot expect that all source compo-

¹A preliminary work (without model definition) of ISA appeared in [93], where the authors searched for fetal ECG (electro-cardiography) subspaces via ICA followed by assigning the estimated ICA elements to different ‘subspaces’ based on domain expert knowledge.

nents are statistically independent. In this case, the independent sources can be multidimensional. For instance, consider the generalization of the cocktail-party problem, where *independent groups* of people are talking about independent topics or more than one group of musicians are playing at the party. The separation task requires an extension of ICA, which is called multidimensional ICA [92], independent subspace analysis (ISA) [107], independent feature subspace analysis [108], subspace ICA [109] or group ICA [110] in the literature. We will use the ISA abbreviation throughout this paper. The several successful applications and the large number of different ISA algorithms show the importance of this field. Successful applications of ISA in signal processing and pattern recognition include: (i) the processing of EEG-fMRI (EEG, electro-encephalography) data [111–113] and natural images [107, 114], (ii) gene expression analysis [115], (iii) learning of face view-subspaces [116], (iv) ECG (electro-cardiography) analysis [92, 109, 110, 117–119], (v) motion segmentation [120], (vi) single-channel source separation [121], (vii) texture classification [122].

We are motivated by:

- a central result of the ICA research, the ISA separation principle.
- the continuously emerging applications using the relaxations of the traditional ICA assumptions.

The ISA Separation Principle. One of the most exciting and fundamental hypotheses of the ICA research is due to Jean-François Cardoso [92], who conjectured that the ISA task can be solved by ICA up to permutation. In other words, it is enough to cluster the ICA elements into statistically dependent groups/subspaces to solve the ISA problem. This principle

- forms the basis of the state-of-the-art ISA solvers. While the extent of this conjecture, the *ISA separation principle* is still an open issue, we have recently shown sufficient conditions for this 10-year-old open question [123].
- enables one to estimate the unknown number and the dimensions of the sources efficiently. Indeed, let us suppose that the dimension of the individual subspaces in ISA is not known. The lack of such knowledge may cause serious

computational burden as one should try all possible

$$D = d_1 + \dots + d_M \quad (d_m > 0, M \leq D) \quad (1.1)$$

dimension allocations (d_m stands for estimation of the m^{th} subspace dimension) for the individual subspaces, where D denotes the total source dimension. The number of these possibilities is given by the so-called partition function $f(D)$, i.e., the number of sets of positive integers that sum up to D . The value of $f(D)$ grows quickly with the argument, its asymptotic behavior is described by the

$$f(D) \sim \frac{e^{\pi\sqrt{2D/3}}}{4D\sqrt{3}}, \quad D \rightarrow \infty \quad (1.2)$$

formula [124, 125]. Making use of the ISA separation principle, however, one can construct large scale ISA algorithms without the prior knowledge of the subspace dimensions by clustering of the ICA elements on the basis of their pairwise mutual information, see, e.g. [126].

- makes it possible to use mature algorithms for the solution of the obtained subproblems, in the example, ICA and clustering methods.

ICA Extensions. Beyond the ISA direction, there exist numerous exciting directions relaxing the traditional assumptions of ICA (one-dimensional sources, i.i.d. sources in time, instantaneous mixture, complete observation), for example:

- Post nonlinear mixture: In this case the linear mixing assumption of ICA is weakened to the composition of a linear and a coordinate-wise acting, so-called post nonlinear (PNL) model. This is the PNL ICA problem [127]. The direction has recently gained widespread attention, for a review see [128].
- Complex valued sources/mixing: In the complex ICA problem, the sources and the mixing process are both realized in the complex domain. The complex-valued computations (i) have been present from the ‘birth’ of ICA [98, 129], (ii) show nice potentials in the analysis of biomedical signals (EEG, fMRI), see e.g., [130–132].

- Incomplete observations: In this case certain parts (coordinates/time instants) of the mixture are not available for observation [133, 134].
- Temporal mixing (convolution): Another extension of the original ICA task is the blind source deconvolution (BSD) problem. Such a problem emerges, for example, at a cocktail party being held in an *echoic* room, and can be modelled by a convolutive mixture relaxing the instantaneous mixing assumption of ICA. For an excellent review on this direction and its applications, see [135].
- Nonparametric dynamics: The general case of sources with unknown, non-parametric dynamics is quite challenging, and very few works focused on this direction [110, 136].

These promising ICA extensions may however often be quite restrictive:

- they usually handle only one type of extensions, e.g.,
 - they allow temporal mixing (BSD), but only for one-dimensional independent sources. Similarly, the available methods for complex and incompletely observable models are only capable of dealing with the simplest ICA model.
 - the current nonparametric techniques focus on
 - * the stationary case / constrained mixing case, and
 - * assume equal and known dimensional hidden independent sources.
- current approaches in the ICA problem family do not allow the application of control/exogenous variables, or active learning of the dynamical systems. The motivation for considering this combination is many-folded. ICA/ISA based models search for hidden variables, but they do not include interaction with environment, i.e., the possibility to apply exogenous variables. *Control assisted data mining* is of particular interest for real world applications. ICA and its extensions have already been successfully applied to certain biomedical data analysis (EEG, ECG, fMRI) problems. The application of control variables in these problems may lead to a new generation of interaction

paradigms. By taking another example, in financial applications, exogenous indicator variables can play the role of control leading to new econometric and financial prediction techniques.

These are the reasons that motivate us to (i) develop novel ISA extensions, ISA based dictionary learning approaches (controlled, incompletely observable, complex, convolutive, nonparametric), where (ii) the dimension of the hidden sources may be not equal/known, and (iii) derive separation principle based solution techniques for the problems. This is the **goal** of Chapter 3.

The paper is structured as follows: In Chapter 2 we focus on (structured) sparse coding schemes, and related kernel based approximation methods. Our novel ISA based dictionary learning approaches are presented in Chapter 3. The efficiency of the structured sparse and ISA based methods are numerically illustrated in Chapter 4 and Chapter 5, respectively. Conclusions are drawn in Chapter 6. Longer technical details and the abbreviations used in the paper are collected in Appendix A. Abbreviations of the paper are listed in Appendix B, see Table B.1.

Notations. Vectors have bold faces (\mathbf{a}), matrices are written by capital letters (\mathbf{A}). Polynomials and $D_1 \times D_2$ sized polynomial matrices are denoted by $\mathbb{R}[z]$ and $\mathbb{R}[z]^{D_1 \times D_2}$, respectively. \Re stands for the real part, \Im for the imaginary part of a complex number. The i^{th} coordinate of vector \mathbf{a} is a_i , $\text{diag}(\mathbf{a})$ denotes the diagonal matrix formed from vector \mathbf{a} . Pointwise product of vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ is denoted by $\mathbf{a} \circ \mathbf{b} = [a_1 b_1; \dots; a_d b_d]$. $\mathbf{b} = [\mathbf{a}_1; \dots; \mathbf{a}_K] \in \mathbb{R}^{d_1 + \dots + d_K}$ denotes the concatenation of vectors $\mathbf{a}_k \in \mathbb{R}^{d_k}$. $\mathbf{A} \otimes \mathbf{B}$ is the Kronecker product of matrices, that is $[a_{ij} \mathbf{B}]$. The uniquely existing Moore-Penrose generalized inverse of matrix $\mathbf{A} \in \mathbb{R}^{D_1 \times D_2}$ is $\mathbf{A}^- \in \mathbb{R}^{D_2 \times D_1}$. For a set (number), $|\cdot|$ denotes the number of elements in the set, (the absolute value of the number). For $\mathbf{a} \in \mathbb{R}^d$, $\mathbf{A} \in \mathbb{R}^{d \times D}$ and for set $O \subseteq \{1, \dots, d\}$, $\mathbf{a}_O \in \mathbb{R}^{|O|}$ denotes the coordinates of vector \mathbf{a} in O , whereas $\mathbf{A}_O \in \mathbb{R}^{|O| \times D}$ contains the rows of matrix \mathbf{A} in O . \mathbf{A}^T is the transposed of matrix \mathbf{A} . \mathbf{A}^* is the adjoint of matrix \mathbf{A} . \mathbf{I} and $\mathbf{0}$ stand for the identity and the null matrices, respectively. $\mathbf{1}$ denotes the vector of only 1s. $\mathcal{O}^D = \{\mathbf{A} \in \mathbb{R}^{D \times D} : \mathbf{A} \mathbf{A}^T = \mathbf{I}\}$ denotes the orthogonal group. $\mathcal{U}^D = \{\mathbf{A} \in \mathbb{C}^{D \times D} : \mathbf{A} \mathbf{A}^* = \mathbf{I}\}$ stands for the unitary group. Operation \max and relations \geq, \leq act component-wise on vectors. The abbreviation $\mathbf{l} \leq \mathbf{x}_1, \dots, \mathbf{x}_N \leq \mathbf{u}$ stands for $\mathbf{l} \leq \mathbf{x}_1 \leq \mathbf{u}, \dots, \mathbf{l} \leq \mathbf{x}_N \leq \mathbf{u}$. For positive numbers p, q , (i) (quasi-)norm ℓ_q of vector $\mathbf{a} \in \mathbb{R}^d$ is $\|\mathbf{a}\|_q = (\sum_{i=1}^d |a_i|^q)^{\frac{1}{q}}$,

(ii) $\ell_{p,q}$ -norm (a.k.a. group norm, mixed ℓ_q/ℓ_p norm) of the same vector is $\|\mathbf{a}\|_{p,q} = \|[\|\mathbf{a}_{P_1}\|_q, \dots, \|\mathbf{a}_{P_K}\|_q]\|_p$, where $\{P_i\}_{i=1}^K$ is a partition of the set $\{1, \dots, d\}$. $S_p^d = \{\mathbf{a} \in \mathbb{R}^d : \|\mathbf{a}\|_p \leq 1\}$ is the unit sphere associated with ℓ_p in \mathbb{R}^d . For any given set system \mathcal{G} , elements of vector $\mathbf{a} \in \mathbb{R}^{|\mathcal{G}|}$ are denoted by a^G , where $G \in \mathcal{G}$, that is $\mathbf{a} = (a^G)_{G \in \mathcal{G}}$. $\Pi_{\mathcal{C}}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{c} \in \mathcal{C}} \|\mathbf{x} - \mathbf{c}\|_2$ denotes the orthogonal projection to the closed and convex set $\mathcal{C} \subseteq \mathbb{R}^d$, where $\mathbf{x} \in \mathbb{R}^d$. Partial derivative of function g with respect to variable \mathbf{x} at point \mathbf{x}_0 is $\frac{\partial g}{\partial \mathbf{x}}(\mathbf{x}_0)$ and $g'(\mathbf{x}_0)$ is the derivative of g at \mathbf{x}_0 . $\mathbb{R}_+^d = \{\mathbf{x} \in \mathbb{R}^d : x_i \geq 0 (\forall i)\}$ stands for the non-negative ortant in \mathbb{R}^d . χ is the characteristic function. The entropy of a random variable is denoted by H , \mathbb{E} is the expectation and $I(\cdot, \dots, \cdot)$ denotes the mutual information of its arguments. $\mathbb{N} = \{0, 1, \dots\}$ stands for the natural numbers. \mathbb{R}_+ and \mathbb{N}_+ denote the set of positive real and the positive natural numbers, respectively. For sets, \times and \setminus stand for direct product and difference, respectively. For $i \leq j$ integers, $[i, j]$ is a shorthand for the interval $\{i, i+1, \dots, j\}$.

Chapter 2

Theory – Group-Structured Dictionary Learning

In this chapter we are dealing with the dictionary learning problem of group-structured sparse codes (Section 2.1) and sparse coding – kernel based approximation equivalences (Section 2.2). We also present a novel, kernel based approximation scheme in Section 2.3, we embed support vector machines to multilayer perceptrons.

2.1 Online Group-Structured Dictionary Learning

In this section, we focus on the problem of online learning of group-structured dictionaries. We define the online group-structured dictionary learning (OSDL) task in Section 2.1.1. Section 2.1.2 is dedicated to our optimization scheme solving the OSDL problem. Numerical examples illustrating the efficiency of our approach are given in Chapter 4.

2.1.1 Problem Definition

We define the online group-structured dictionary learning (OSDL) task [137, 138] as follows. Let the dimension of our observations be denoted by d_x . Assume that in each time instant ($i = 1, 2, \dots$) a set $O_i \subseteq \{1, \dots, d_x\}$ is given, that is, we know which coordinates are observable at time i , and our observation is \mathbf{x}_{O_i} . We aim to find a dictionary $\mathbf{D} \in \mathbb{R}^{d_x \times d_\alpha}$ that can approximate the observations \mathbf{x}_{O_i}

well from the linear combination of its columns. We assume that the columns of \mathbf{D} belong to a closed, convex, and bounded set $\mathcal{D} = \times_{i=1}^{d_\alpha} \mathcal{D}_i$. To formulate the cost of dictionary \mathbf{D} , we first consider a *fixed* time instant i , observation \mathbf{x}_{O_i} , dictionary \mathbf{D} , and define the hidden representation α_i associated to this triple $(\mathbf{x}_{O_i}, \mathbf{D}, O_i)$. Representation α_i is allowed to belong to a closed, convex set $\mathcal{A} \subseteq \mathbb{R}^{d_\alpha}$ ($\alpha_i \in \mathcal{A}$) with certain structural constraints. We express the structural constraint on α_i by making use of a given \mathcal{G} group structure, which is a set system (also called hypergraph) on $\{1, \dots, d_\alpha\}$. We also assume that a set of linear transformations $\{\mathbf{A}^G \in \mathbb{R}^{d_G \times d_\alpha}\}_{G \in \mathcal{G}}$ is given for us. We will use them as parameters to define the structured regularization on the codes. Representation α belonging to a triple $(\mathbf{x}_O, \mathbf{D}, O)$ is defined as the solution of the structured sparse coding task

$$l(\mathbf{x}_O, \mathbf{D}_O) = l_{\mathcal{A}, \kappa, \mathcal{G}, \{\mathbf{A}^G\}_{G \in \mathcal{G}}, \eta}(\mathbf{x}_O, \mathbf{D}_O) \quad (2.1)$$

$$= \min_{\alpha \in \mathcal{A}} \left[\frac{1}{2} \|\mathbf{x}_O - \mathbf{D}_O \alpha\|_2^2 + \kappa \Omega(\alpha) \right], \quad (2.2)$$

where $l(\mathbf{x}_O, \mathbf{D}_O)$ denotes the loss, $\kappa > 0$, and

$$\Omega(\mathbf{y}) = \Omega_{\mathcal{G}, \{\mathbf{A}^G\}_{G \in \mathcal{G}}, \eta}(\mathbf{y}) = \|(\|\mathbf{A}^G \mathbf{y}\|_2)_{G \in \mathcal{G}}\|_\eta \quad (2.3)$$

is the group structure inducing regularizer associated to \mathcal{G} and $\{\mathbf{A}^G\}_{G \in \mathcal{G}}$, and $\eta \in (0, 2)$. Here, the first term of (2.2) is responsible for the quality of approximation on the observed coordinates, and (2.3) performs regularization defined by the group structure/hypergraph \mathcal{G} and the $\{\mathbf{A}^G\}_{G \in \mathcal{G}}$ linear transformations. The OSDL problem is defined as the minimization of the cost function:

$$\min_{\mathbf{D} \in \mathcal{D}} f_t(\mathbf{D}) := \frac{1}{\sum_{j=1}^t (j/t)^\rho} \sum_{i=1}^t \left(\frac{j}{t} \right)^\rho l(\mathbf{x}_{O_i}, \mathbf{D}_{O_i}), \quad (2.4)$$

that is, we aim to minimize the average loss of the dictionary, where ρ is a non-negative forgetting rate. If $\rho = 0$, the classical average

$$f_t(\mathbf{D}) = \frac{1}{t} \sum_{i=1}^t l(\mathbf{x}_{O_i}, \mathbf{D}_{O_i}) \quad (2.5)$$

is obtained. When $\eta \leq 1$, then for a code vector α , the regularizer Ω aims at eliminating the $\mathbf{A}^G \alpha$ terms ($G \in \mathcal{G}$) by making use of the sparsity-inducing property of the $\|\cdot\|_\eta$ norm [63]. For $O_i = \{1, \dots, d_x\}$ ($\forall i$), we get the fully observed OSDL task.

Below we list a few special cases of the OSDL problem:

- Special cases for \mathcal{G} :
 - If $|\mathcal{G}| = d_\alpha$ and $\mathcal{G} = \{\{1\}, \{2\}, \dots, \{d_\alpha\}\}$, then no dependence is assumed between coordinates α_i , and the problem reduces to the classical task of learning ‘dictionaries with sparse codes’ [139].
 - If $|\mathcal{G}| = d_\alpha$ and $\mathcal{G} = \{desc_1, \dots, desc_{d_\alpha}\}$, where $desc_i$ stands for the i^{th} node (α_i) of a tree and its descendants, then we have a tree-structured, hierarchical representation [62].
 - If $|\mathcal{G}| = d_\alpha$, and $\mathcal{G} = \{NN_1, \dots, NN_{d_\alpha}\}$, where NN_i denotes the neighbors of the i^{th} point (α_i) in radius r on a grid, then we obtain a grid representation [66].
 - If $\mathcal{G} = \{\{1\}, \dots, \{d_\alpha\}, \{1, \dots, d_\alpha\}\}$, then we have an elastic net representation [140].
 - $\mathcal{G} = \{\{[1, k]\}_{k \in \{1, \dots, d_\alpha - 1\}}, \{[k, d_\alpha]\}_{k \in \{2, \dots, d_\alpha\}}\}$ intervals lead to a 1D contiguous, nonzero representation. One can also generalize the construction to higher dimensions [49].
 - If \mathcal{G} is a partition of $\{1, \dots, d_\alpha\}$, then non-overlapping group structure is obtained. In this case, we are working with block-sparse (a.k.a. group Lasso) representation [6].
- Special cases for $\{\mathbf{A}^G\}_{G \in \mathcal{G}}$:
 - Let (V, E) be a given graph, where V and E denote the set of nodes and edges, respectively. For each $e = (i, j) \in E$, we also introduce (w_{ij}, v_{ij}) weight pairs. Now, if we set

$$\Omega(\mathbf{y}) = \sum_{e=(i,j) \in E: i < j} w_{ij} |y_i - v_{ij} y_j|, \quad (2.6)$$

then we obtain the graph-guided fusion penalty [16]. The groups $G \in \mathcal{G}$ correspond to the (i, j) pairs, and in this case

$$\mathbf{A}^G = [w_{ij}, -w_{ij}v_{ij}] \in \mathbb{R}^{1 \times 2}. \quad (2.7)$$

As a special case, for a chain graph we get the standard fused Lasso penalty by setting the weights to one [141]:

$$\Omega(\mathbf{y}) = FL(\mathbf{y}) = \sum_{j=1}^{d_\alpha-1} |y_{j+1} - y_j|. \quad (2.8)$$

- The fused Lasso penalty can be seen as a zero-order difference approach. One can also take first order

$$\Omega(\mathbf{y}) = \sum_{j=2}^{d_\alpha-1} |-y_{j-1} + 2y_j - y_{j+1}| \quad (2.9)$$

differences arriving to linear trend filtering (also called ℓ_1 trend filtering) [142], or its higher order variants lead to polynomial filtering techniques.

- By restricting the \mathcal{G} group structure to have a single element ($|\mathcal{G}| = 1$) and η to 1, we obtain the

$$\Omega(\mathbf{y}) = \|\mathbf{A}\mathbf{y}\|_1 \quad (2.10)$$

generalized Lasso penalty [43, 59].

- Let $\nabla \mathbf{y} \in \mathbb{R}^{d_1 \times d_2}$ denote the discrete differential of an image $\mathbf{y} \in \mathbb{R}^{d_1 \times d_2}$ at position $(i, j) \in \{1, \dots, d_1\} \times \{1, \dots, d_2\}$:

$$(\nabla \mathbf{y})_{ij} = [(\nabla \mathbf{y})_{ij}^1; (\nabla \mathbf{y})_{ij}^2], \quad (2.11)$$

where

$$(\nabla \mathbf{y})_{ij}^1 = (y_{i+1,j} - y_{i,j}) \chi_{\{i < d_1\}}, \quad (2.12)$$

$$(\nabla \mathbf{y})_{ij}^2 = (y_{i,j+1} - y_{i,j}) \chi_{\{j < d_2\}}. \quad (2.13)$$

Using these notations, the total variation of \mathbf{y} is defined as follows [143]:

$$\Omega(\mathbf{y}) = \|\mathbf{y}\|_{TV} = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \|(\nabla \mathbf{y})_{ij}\|_2. \quad (2.14)$$

• Special cases for \mathcal{D}, \mathcal{A} :

- $\mathcal{D}_i = S_2^{d_x} (\forall i), \mathcal{A} = \mathbb{R}^{d_\alpha}$: columns of dictionary \mathbf{D} are constrained to be in the Euclidean unit sphere.
- $\mathcal{D}_i = S_2^{d_x} \cap \mathbb{R}_+^{d_x} (\forall i), \mathcal{A} = \mathbb{R}_+^{d_\alpha}$: This is the structured non-negative matrix factorization (NMF) problem.
- $\mathcal{D}_i = S_1^{d_x} \cap \mathbb{R}_+^{d_x} (\forall i), \mathcal{A} = \mathbb{R}_+^{d_\alpha}$: This is the structured mixture-of-topics problem.
- Beyond $\mathbb{R}^d, S_1^d, S_2^d, S_1^d \cap \mathbb{R}_+^d$, and $S_2^d \cap \mathbb{R}_+^d$, several other constraints can also be motivated for \mathcal{D}_i and \mathcal{A} . In the above mentioned examples, the group-norm, elastic net, and fused Lasso constraints have been applied in a ‘soft’ manner, with the help of the Ω regularization. However, we can enforce these constraints in a ‘hard’ way as well: During optimization (Section 2.1.2), we can exploit the fact that the projection to the \mathcal{D}_i and \mathcal{A} constraint sets can be computed efficiently. Such constraint sets include [77, 144, 145], e.g., the
 - * $\{\mathbf{c} : \|\mathbf{c}\|_{p,q} \leq 1\}$ group norms,
 - * $\{\mathbf{c} : \gamma_1 \|\mathbf{c}\|_1 + \gamma_2 \|\mathbf{c}\|_2^2 \leq 1\}$ elastic net, and
 - * $\{\mathbf{c} : \gamma_1 \|\mathbf{c}\|_1 + \gamma_2 \|\mathbf{c}\|_2^2 + \gamma_3 FL(\mathbf{c}) \leq 1\}$ fused Lasso ($\gamma_1, \gamma_2, \gamma_3 > 0$).
- When applying group norms for both the codes α and the dictionary \mathbf{D} , we arrive at a double structured dictionary learning scheme.

In sum, the OSDL model provides a unified dictionary learning framework for

several actively studied structured sparse coding problems, naturally extends them for incomplete observations, and allows non-convex regularization as well.

2.1.2 Optimization

We consider the optimization of cost function (2.4), which is equivalent to the joined optimization of dictionary \mathbf{D} and coefficients $\{\alpha_i\}_{i=1}^t$:

$$\arg \min_{\mathbf{D} \in \mathcal{D}, \{\alpha_i \in \mathcal{A}\}_{i=1}^t} f_t(\mathbf{D}, \{\alpha_i\}_{i=1}^t), \quad (2.15)$$

where

$$f_t = \frac{1}{\sum_{j=1}^t (j/t)^\rho} \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \left[\frac{1}{2} \|\mathbf{x}_{O_i} - \mathbf{D}_{O_i} \alpha_i\|_2^2 + \kappa \Omega(\alpha_i) \right]. \quad (2.16)$$

Assume that our samples \mathbf{x}_i are emitted from an i.i.d. source $p(\mathbf{x})$, and we can observe \mathbf{x}_{O_i} . We execute the online optimization of dictionary \mathbf{D} (i.e., the minimization of (2.16)) through alternations:

1. For the actual sample \mathbf{x}_{O_t} we optimize hidden representation α_t belonging to \mathbf{x}_{O_t} using our estimated dictionary \mathbf{D}_{t-1} and solving the minimization task

$$\alpha_t = \underset{\alpha \in \mathcal{A}}{\operatorname{argmin}} \left[\frac{1}{2} \|\mathbf{x}_{O_t} - (\mathbf{D}_{t-1})_{O_t} \alpha\|_2^2 + \kappa \Omega(\alpha) \right]. \quad (2.17)$$

2. We use hidden representations $\{\alpha_i\}_{i=1}^t$ and update \mathbf{D}_{t-1} by means of quadratic optimization

$$\hat{f}_t(\mathbf{D}_t) = \min_{\mathbf{D} \in \mathcal{D}} f_t(\mathbf{D}, \{\alpha_i\}_{i=1}^t). \quad (2.18)$$

In the next subsections, we elaborate on the optimization of representation α in (2.17) and the dictionary \mathbf{D} in (2.18).

Representation update (α)

Objective function (2.17) is not convex in α . We use a variational method to find a solution: (i) we rewrite the term Ω by introducing an auxiliary variable (z) that

converts the expression to a quadratic one in α , and then (ii) we use an explicit solution to \mathbf{z} and continue by iteration. Namely, we use Lemma 3.1 of [63]: for any $\mathbf{y} \in \mathbb{R}^d$ and $\eta \in (0, 2)$

$$\|\mathbf{y}\|_\eta = \min_{\mathbf{z} \in \mathbb{R}_+^d} \left[\frac{1}{2} \sum_{i=1}^d \frac{y_i^2}{z_i} + \frac{1}{2} \|\mathbf{z}\|_\beta \right], \quad (2.19)$$

where $\beta = \frac{\eta}{2-\eta}$, and it takes its minimum value at

$$z_i^* = |y_i|^{2-\eta} \|\mathbf{y}\|_\eta^{\eta-1}. \quad (2.20)$$

We apply this relation for the term Ω in (2.17) (see Eq. (2.3)), and have that

$$2\Omega(\alpha) = \min_{\mathbf{z}=[(z^G)_{G \in \mathcal{G}}] \in \mathbb{R}_+^{|\mathcal{G}|}} \left[\sum_{G \in \mathcal{G}} \frac{\|\mathbf{A}^G \alpha\|_2^2}{z^G} + \|\mathbf{z}\|_\beta \right] \quad (2.21)$$

$$= \min_{\mathbf{z} \in \mathbb{R}_+^{|\mathcal{G}|}} \left[\alpha^T \mathbf{H} \alpha + \|\mathbf{z}\|_\beta \right], \quad (2.22)$$

where

$$\mathbf{H} = \mathbf{H}(\mathbf{z}) = \sum_{G \in \mathcal{G}} (\mathbf{A}^G)^T \mathbf{A}^G / z^G. \quad (2.23)$$

Inserting (2.22) into (2.17) we get the optimization task:

$$\arg \min_{\alpha \in \mathcal{A}, \mathbf{z} \in \mathbb{R}_+^{|\mathcal{G}|}} J(\alpha, \mathbf{z}) = \frac{1}{2} \|\mathbf{x}_{O_t} - (\mathbf{D}_{t-1})_{O_t} \alpha\|_2^2 + \kappa \frac{1}{2} \left(\alpha^T \mathbf{H} \alpha + \|\mathbf{z}\|_\beta \right). \quad (2.24)$$

One can solve the minimization of $J(\alpha, \mathbf{z})$ by alternations:

1. For given \mathbf{z} : we can use least mean square solver for α when $\mathcal{A} = \mathbb{R}^{d_\alpha}$ in (2.24), and non-negative least square solver when $\mathcal{A} = \mathbb{R}_+^{d_\alpha}$. For the general case, the cost function $J(\alpha, \mathbf{z})$ is quadratic in α and is subject to convex and closed constraints ($\alpha \in \mathcal{A}$). There are standard solvers for this case [146, 147], too.
2. For given α : According to (2.19), the minimum $\mathbf{z} = (z^G)_{G \in \mathcal{G}}$ can be found

as

$$z^G = \|\mathbf{A}^G \boldsymbol{\alpha}\|_2^{2-\eta} \|(\|\mathbf{A}^G \boldsymbol{\alpha}\|_2)_{G \in \mathcal{G}}\|_\eta^{\eta-1}. \quad (2.25)$$

Note that for numerical stability smoothing, $\mathbf{z} = \max(\mathbf{z}, \varepsilon)$ ($0 < \varepsilon \ll 1$), is suggested in practice.

Dictionary update (D)

We use block-coordinate descent (BCD) [147] for the optimization of (2.18). This optimization is not influenced by the regularizer $\Omega(\boldsymbol{\alpha})$, since it is independent of \mathbf{D} . Thus the task (2.18) is *similar* to the fully observable case [77], where for $O_i = \{1, \dots, d_x\}$ ($\forall i$) it has been shown that the BCD method can work without storing all of the vectors $\mathbf{x}_i, \boldsymbol{\alpha}_i$ ($i \leq t$). Instead, it is sufficient to keep certain statistics that characterize \hat{f}_t , which can be updated online. This way, optimization of \hat{f}_t in (2.18) becomes online, too. As it will be elaborated below, (i) certain statistics describing \hat{f}_t can also be derived for the partially observed case, which (ii) can be updated online with a single exception, and (iii) a good approximation exists for that exception (see Chapter 4).

During the BCD optimization, columns of \mathbf{D} are minimized sequentially: other columns than the actually updated \mathbf{d}_j (i.e., $\mathbf{d}_i, i \neq j$) are kept fixed. The function \hat{f}_t is quadratic in \mathbf{d}_j . During minimization we search for its minimum (denoted by \mathbf{u}_j) and project the result to the constraint set \mathcal{D}_j ($\mathbf{d}_j \leftarrow \Pi_{\mathcal{D}_j}(\mathbf{u}_j)$). To find this \mathbf{u}_j , we solve the equation $\frac{\partial \hat{f}_t}{\partial \mathbf{d}_j}(\mathbf{u}_j) = \mathbf{0}$, which leads (as we show it in Appendix A.1.1-A.1.2) to the following linear equation system

$$\mathbf{C}_{j,t} \mathbf{u}_j = \mathbf{b}_{j,t} - \mathbf{e}_{j,t} + \mathbf{C}_{j,t} \mathbf{d}_j, \quad (2.26)$$

where $\mathbf{C}_{j,t} \in \mathbb{R}^{d_x \times d_x}$ is a diagonal coefficient matrix, and

$$\mathbf{C}_{j,t} = \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \Delta_i \alpha_{i,j}^2, \quad (2.27)$$

$$\mathbf{B}_t = \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \Delta_i \mathbf{x}_i \alpha_i^T = [\mathbf{b}_{1,t}, \dots, \mathbf{b}_{d_\alpha,t}], \quad (2.28)$$

$$\mathbf{e}_{j,t} = \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \Delta_i \mathbf{D} \alpha_i \alpha_{i,j}. \quad (2.29)$$

Here Δ_i represents a diagonal matrix corresponding to O_i (element j in the diagonal is 1 if $j \in O_i$, and 0 otherwise). $\mathbf{C}_{j,t} \in \mathbb{R}^{d_x \times d_x}$ and $\mathbf{B}_t \in \mathbb{R}^{d_x \times d_\alpha}$ take the form of

$$\mathbf{M}_t = \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \mathbf{N}_i \quad (2.30)$$

matrix series/statistics, and thus (as we detail it in Appendix A.1.1-A.1.2) they can be updated as

$$\mathbf{C}_{j,t} = \gamma_t \mathbf{C}_{j,t-1} + \Delta_t \alpha_{tj}^2, \quad \mathbf{B}_t = \gamma_t \mathbf{B}_{t-1} + \Delta_t \mathbf{x}_t \alpha_t^T, \quad (2.31)$$

with initialization $\mathbf{C}_{j,0} = \mathbf{0}$, $\mathbf{B}_0 = \mathbf{0}$ for the case of $\rho = 0$, and with arbitrary initialization for $\rho > 0$, where $\gamma_t = (1 - \frac{1}{t})^\rho$. For the fully observed case ($\Delta_i = \mathbf{I}$, $\forall i$), one can pull out \mathbf{D} from $\mathbf{e}_{j,t} \in \mathbb{R}^{d_x}$, the remaining part is of the form \mathbf{M}_t , and thus it can be updated online giving rise to the update rules in [77], see Appendix A.1.1-A.1.2. In the general case this procedure cannot be applied (matrix \mathbf{D} changes during the BCD updates). According to our numerical experiences (see Chapter 4) an efficient online approximation for $\mathbf{e}_{j,t}$ is

$$\mathbf{e}_{j,t} = \gamma_t \mathbf{e}_{j,t-1} + \Delta_t \mathbf{D}_t \alpha_t \alpha_{t,j}, \quad (2.32)$$

with the actual estimation for \mathbf{D}_t and with initialization $\mathbf{e}_{j,0} = \mathbf{0}$ ($\forall j$). We note that

1. convergence is often speeded up if the updates of statistics

$$\{\{\mathbf{C}_{j,t}\}_{j=1}^{d_\alpha}, \mathbf{B}_t, \{\mathbf{e}_{j,t}\}_{j=1}^{d_\alpha}\} \quad (2.33)$$

are made in batches of R samples $\mathbf{x}_{O_{t,1}}, \dots, \mathbf{x}_{O_{t,R}}$ (in R -tuple mini-batches). The pseudocode of the OSDL method with mini-batches is presented in Table 2.1-2.3. Table 2.2 calculates the representation for a fixed dictionary, and Table 2.3 learns the dictionary using fixed representations. Table 2.1 invokes both of these subroutines.

2. The trick in the representation update was that the auxiliary variable \mathbf{z} ‘replaced’ the Ω term with a quadratic one in α . One could use further $g(\alpha)$ regularizers augmenting Ω in (2.16) provided that the corresponding $J(\alpha, \mathbf{z}) + g(\alpha)$ cost function (see Eq. (2.24)) can be efficiently optimized in $\alpha \in \mathcal{A}$.

2.2 Generalized Support Vector Machines and ϵ -Sparse Representations

In this section we present an extension of sparse coding in RKHSs, and show its equivalence to a generalized family of SVMs. The structure of the section is as follows: we briefly summarize the basic properties that will be used throughout the section of kernels with the associated notion of RKHSs and SVMs in Section 2.2.1 and Section 2.2.2, respectively. In Section 2.2.3 we present our equivalence result.

Let us assume that we are given $\{(\mathbf{x}_i, y_i)\}_{i=1}^I$ input-output sample pairs, where $\mathbf{x}_i \in \mathcal{X}$ (input space) and $y_i \in \mathbb{R}$. Our goal is to approximate the $\mathbf{x} \mapsto y$ relation. One can choose the approximating function from different function classes. In the sequel, we will focus on approximations, where this function class is a so-called reproducing kernel Hilbert space.

2.2.1 Reproducing Kernel Hilbert Space

Below, we briefly summarize the concepts of kernel, feature map, feature space, reproducing kernel, reproducing kernel Hilbert space and Gram matrix.

Let \mathcal{X} be non-empty set. Then a function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is called a *kernel* on \mathcal{X} if there exists a Hilbert space \mathcal{H} and a map $\varphi : \mathcal{X} \mapsto \mathcal{H}$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ we have

$$k(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{H}}. \quad (2.34)$$

Table 2.1: Pseudocode: Online Group-Structured Dictionary Learning.

Algorithm (Online Group-Structured Dictionary Learning)
<p>Input of the algorithm</p> <p>$\mathbf{x}_{t,r} \sim p(\mathbf{x})$, (observation: $\mathbf{x}_{O_{t,r}}$, observed positions: $O_{t,r}$), T (number of mini-batches), R (size of the mini-batches), \mathcal{G} (group structure), ρ (≥ 0 forgetting factor), κ (> 0 tradeoff-), η ($\in (0, 2)$ regularization constant), $\{\mathbf{A}^G\}_{G \in \mathcal{G}}$ (linear transformations), \mathcal{A} (constraint set for α), \mathbf{D}_0 (initial dictionary), $\mathcal{D} = \times_{i=1}^{d_\alpha} \mathcal{D}_i$ (constraint set for \mathbf{D}) inner loop constants: ϵ (smoothing), T_α, T_D (number of iterations).</p> <p>Initialization</p> <p>$\mathbf{C}_{j,0} = \mathbf{0} \in \mathbb{R}^{d_x}$, $\mathbf{e}_{j,0} = \mathbf{0} \in \mathbb{R}^{d_x}$ ($j = 1, \dots, d_\alpha$), $\mathbf{B}_0 = \mathbf{0} \in \mathbb{R}^{d_x \times d_\alpha}$.</p> <p>Optimization</p> <p>for $t = 1 : T$</p> <p> Draw samples for mini-batch from $p(\mathbf{x})$: $\{\mathbf{x}_{O_{t,1}}, \dots, \mathbf{x}_{O_{t,R}}\}$.</p> <p> Compute the $\{\alpha_{t,1}, \dots, \alpha_{t,R}\}$ representations: $\alpha_{t,r} = \text{Representation}(\mathbf{x}_{O_{t,r}}, (\mathbf{D}_{t-1})_{O_{t,r}}, \mathcal{G}, \{\mathbf{A}^G\}_{G \in \mathcal{G}}, \kappa, \eta, \mathcal{A}, \epsilon, T_\alpha)$, $(r = 1, \dots, R)$.</p> <p> Update the statistics of the cost function: $\gamma_t = (1 - \frac{1}{t})^\rho$, $\mathbf{C}_{j,t} = \gamma_t \mathbf{C}_{j,t-1} + \frac{1}{R} \sum_{r=1}^R \Delta_{t,r} \alpha_{t,r,j}^2$, $j = 1, \dots, d_\alpha$, $\mathbf{B}_t = \gamma_t \mathbf{B}_{t-1} + \frac{1}{R} \sum_{r=1}^R \Delta_{t,r} \mathbf{x}_{t,r} \alpha_{t,r}^T$, $\mathbf{e}_{j,t} = \gamma_t \mathbf{e}_{j,t-1}$, $j = 1, \dots, d_\alpha$. % (part-1)</p> <p> Compute \mathbf{D}_t using BCD: $\mathbf{D}_t = \text{Dictionary}(\{\mathbf{C}_{j,t}\}_{j=1}^{d_\alpha}, \mathbf{B}_t, \{\mathbf{e}_{j,t}\}_{j=1}^{d_\alpha}, \mathcal{D}, T_D, \{O_{t,r}\}_{r=1}^R, \{\alpha_{t,r}\}_{r=1}^R)$.</p> <p> Finish the update of $\{\mathbf{e}_{j,t}\}_{j=1}^{d_\alpha}$-s: % (part-2) $\mathbf{e}_{j,t} = \mathbf{e}_{j,t} + \frac{1}{R} \sum_{r=1}^R \Delta_{t,r} \mathbf{D}_t \alpha_{t,r} \alpha_{t,r,j}$, $j = 1, \dots, d_\alpha$.</p> <p>end</p> <p>Output of the algorithm</p> <p>\mathbf{D}_T (learned dictionary).</p>

Table 2.2: Pseudocode for *representation* estimation using fixed dictionary.

Algorithm (Representation)
<p>Input of the algorithm \mathbf{x} (observation), $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_{d_\alpha}]$ (dictionary), \mathcal{G} (group structure), $\{\mathbf{A}^G\}_{G \in \mathcal{G}}$ (linear transformations), κ (tradeoff-), η (regularization constant), \mathcal{A} (constraint set for α), ϵ (smoothing), T_α (number of iterations).</p> <p>Initialization $\alpha \in \mathbb{R}^{d_\alpha}$.</p> <p>Optimization for $t = 1 : T_\alpha$ Compute \mathbf{z}: $z^G = \max \left(\ \mathbf{A}^G \alpha\ _2^{2-\eta} \left\ (\ \mathbf{A}^G \alpha\ _2)_{G \in \mathcal{G}} \right\ _\eta^{\eta-1}, \epsilon \right), G \in \mathcal{G}$. Compute α: compute \mathbf{H}: $\mathbf{H} = \sum_{G \in \mathcal{G}} (\mathbf{A}^G)^T \mathbf{A}^G / z^G$, $\alpha = \underset{\alpha \in \mathcal{A}}{\operatorname{argmin}} [\ \mathbf{x} - \mathbf{D}\alpha\ _2^2 + \kappa \alpha^T \mathbf{H} \alpha]$. end</p> <p>Output of the algorithm α (estimated representation).</p>

We call φ a *feature map* and \mathcal{H} a feature space of k . Given a kernel neither the feature map, nor the feature space are uniquely determined. However, one can always construct a canonical feature space, namely the reproducing kernel Hilbert space (RKHS) [82]. Let us now recall the basic theory of these spaces.

Let \mathcal{X} be non-empty set, and \mathcal{H} be a Hilbert space over \mathcal{X} , i.e., a Hilbert space which consists of functions mapping from \mathcal{X} .

- The space \mathcal{H} is called a *RKHS* over \mathcal{X} if for all $\mathbf{x} \in \mathcal{X}$ the Dirac functional $\delta_{\mathbf{x}} : \mathcal{H} \mapsto \mathbb{R}$ defined by $\delta_{\mathbf{x}}(f) = f(\mathbf{x}), f \in \mathcal{H}$, is continuous.
- A function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is called a reproducing kernel of \mathcal{H} if we have $k(\cdot, \mathbf{x}) \in \mathcal{H}$ for all $\mathbf{x} \in \mathcal{X}$ and the *reproducing property*

$$f(\mathbf{x}) = \langle f(\cdot), k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}, \quad (2.35)$$

holds for all $\mathbf{x} \in \mathcal{X}$ and $f \in \mathcal{H}$.

The reproducing kernels are kernels in the sense of (2.34) since $\varphi : \mathcal{X} \mapsto \mathcal{H}$ defined

Table 2.3: Pseudocode for *dictionary* estimation using fixed representations.

Algorithm (Dictionary)
<p>Input of the algorithm</p> <p>$\{\mathbf{C}_j\}_{j=1}^{d_\alpha}$, $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_{d_\alpha}]$, $\{\mathbf{e}_j\}_{j=1}^{d_\alpha}$ (statistics of the cost function), $\mathcal{D} = \times_{i=1}^{d_\alpha} \mathcal{D}_i$ (constraint set for \mathbf{D}), T_D (number of \mathbf{D} iterations), $\{O_r\}_{r=1}^R$ (equivalent to $\{\Delta_r\}_{r=1}^R$), $\{\alpha_r\}_{r=1}^R$ (observed positions, estimated representations).</p> <p>Initialization</p> <p>$\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_{d_\alpha}]$.</p> <p>Optimization</p> <p>for $t = 1 : T_D$</p> <p> for $j = 1 : d_\alpha$ %update the j^{th} column of \mathbf{D}</p> <p> Compute $\{\mathbf{e}_j\}_{j=1}^{d_\alpha}$’-s:</p> <p> $\mathbf{e}_j^{temp} = \mathbf{e}_j + \frac{1}{R} \sum_{r=1}^R \Delta_r \mathbf{D} \alpha_r \alpha_{r,j}$.</p> <p> Compute \mathbf{u}_j solving the linear equation system:</p> <p> $\mathbf{C}_j \mathbf{u}_j = \mathbf{b}_j - \mathbf{e}_j^{temp} + \mathbf{C}_j \mathbf{d}_j$.</p> <p> Project \mathbf{u}_j to the constraint set:</p> <p> $\mathbf{d}_j = \Pi_{\mathcal{D}_j}(\mathbf{u}_j)$.</p> <p> end</p> <p>end</p> <p>Output of the algorithm</p> <p>\mathbf{D} (estimated dictionary).</p>

by $\varphi(\mathbf{x}) = k(\cdot, \mathbf{x})$ is a feature map of k . A RKHS space can be uniquely identified by its k reproducing kernel, hence in the sequel we will use the notation $\mathcal{H} = \mathcal{H}(k)$. The *Gram matrix* of k on the point set $\{\mathbf{x}_1, \dots, \mathbf{x}_l\}$ ($\mathbf{x}_i \in \mathcal{X}, \forall i$) is defined as

$$\mathbf{G} = [G_{ij}]_{i,j=1}^l = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^l. \quad (2.36)$$

An important property of RKHSs, is that the scalar products in the feature space can be computed implicitly by means of the kernel. Indeed, let us suppose that $\mathbf{w} \in \mathcal{H} = \mathcal{H}(k)$ has an *expansion* of the form

$$\mathbf{w} = \sum_{j=1}^N \alpha_j \varphi(\mathbf{z}_j), \quad (2.37)$$

Table 2.4: Kernel examples.

Name	Kernel (k)	Assumption
linear kernel	$k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$	
RBF ^a kernel	$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\ \mathbf{x}-\mathbf{y}\ ^2}{2\sigma^2}}$	
Mahalanobis kernel	$k(\mathbf{x}, \mathbf{y}) = e^{-(\mathbf{x}-\mathbf{y})^T \Sigma^{-1}(\mathbf{x}-\mathbf{y})}$	$\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_d^2)$
polynomial kernel	$k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^p$	$p \in \mathbb{N}_+$
complete polynomial kernel	$k(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^p$	$p \in \mathbb{N}_+, c \in \mathbb{R}_+$
Dirichlet kernel	$k(x, y) = \frac{\sin((N+\frac{1}{2})(x-y))}{\sin(\frac{x-y}{2})}$	$N \in \mathbb{N}$

^aRBF stands for radial basis function.

where $\alpha_j \in \mathbb{R}$ and $\mathbf{z}_j \in \mathcal{X}$. Then

$$f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}} = \left\langle \sum_{j=1}^N \alpha_j \boldsymbol{\varphi}(\mathbf{z}_j), \boldsymbol{\varphi}(\mathbf{x}) \right\rangle_{\mathcal{H}} \quad (2.38)$$

$$= \sum_{j=1}^N \alpha_j \langle \boldsymbol{\varphi}(\mathbf{z}_j), \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}} = \sum_{j=1}^N \alpha_j k(\mathbf{z}_j, \mathbf{x}), \quad (2.39)$$

i.e., function $f_{\mathbf{w}}$ can be evaluated by means of coefficients α_j , samples \mathbf{z}_j and the kernel k without explicit reference to representation $\boldsymbol{\varphi}(\mathbf{x})$. This technique is called the *kernel trick*. In Table 2.4 we list some well-known kernels.

2.2.2 Support Vector Machine

Now, we present the concept of support vector machines (SVM). In the SVM framework the approximating function for the $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ samples are based on a $\mathcal{H} = \mathcal{H}(k)$ RKHS, and takes the form

$$f_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}} + b, \quad (2.40)$$

Although this function $f_{\mathbf{w},b}$ is nonlinear as an $\mathcal{X} \mapsto \mathbb{R}$ mapping, it is a linear function of the feature representation $\boldsymbol{\varphi}(\mathbf{x})$. For different choices of RKHS \mathcal{H} , $f_{\mathbf{w},b}$ may realize, e.g., polynomial, Fourier, or even infinite dimensional feature representations.

The cost function of the SVM regression is

$$H(\mathbf{w}, b) = C \sum_{i=1}^l |y_i - f_{\mathbf{w},b}(\mathbf{x}_i)|_\epsilon + \frac{1}{2} \|\mathbf{w}\|_{\mathcal{H}}^2 \rightarrow \min_{\mathbf{w} \in \mathcal{H}, b \in \mathbb{R}}, \quad (2.41)$$

where $C > 0$ and

$$|r|_\epsilon = \{0, \text{ if } |r| \leq \epsilon; |r| - \epsilon \text{ otherwise}\} \quad (2.42)$$

is the ϵ -insensitive cost. In (2.41), the first term is responsible for the quality of approximation on the sample points $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ in ϵ -insensitive sense; the second term corresponds to a regularization by the $\|\mathbf{w}\|_{\mathcal{H}}^2 = \langle \mathbf{w}, \mathbf{w} \rangle_{\mathcal{H}}$ squared norm, and C balances between the two terms.

Exploiting the special form of the SVM cost (2.41) and the representation theorem in RKHSs [148], the optimization can be executed and function $f_{\mathbf{w},b}$ can be computed (even for infinite dimensional feature representations) by solving the dual of (2.41), a quadratic programming (QP) problem, which takes the form [87]

$$\begin{aligned} \frac{1}{2} (\mathbf{d}^* - \mathbf{d})^T \mathbf{G} (\mathbf{d}^* - \mathbf{d}) - (\mathbf{d}^* - \mathbf{d})^T \mathbf{y} + (\mathbf{d}^* + \mathbf{d})^T \mathbf{e} \mathbf{1} &\rightarrow \min_{\mathbf{d}^* \in \mathbb{R}^l, \mathbf{d} \in \mathbb{R}^l}, \\ \text{subject to } \begin{cases} C \mathbf{1} \geq \mathbf{d}^*, \mathbf{d} &\geq \mathbf{0} \\ (\mathbf{d}^* - \mathbf{d})^T \mathbf{1} &= 0 \end{cases}, \end{aligned} \quad (2.43)$$

where $\mathbf{G} = [G_{ij}]_{i,j=1}^l = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^l$ is the Gram matrix of the $\{\mathbf{x}_i\}_{i=1}^l$ samples.

2.2.3 Equivalence of Generalized Support Vector Machines and ϵ -Sparse Coding

Having the notions of SVM and RKHS at hand, we are now able to focus on sparse coding problems in RKHSs. Again, it is assumed that we are given l samples $(\{(\mathbf{x}_i, y_i)\}_{i=1}^l)$. First, we focus on the noiseless case, i.e., it is assumed that $f(\mathbf{x}_i) = y_i$ ($\forall i$) for a suitable $f \in \mathcal{H}$. In the noiseless case, [83] has recently

formulated a sparse coding problem in RKHSs as the optimization problem

$$\frac{1}{2} \left\| f(\cdot) - \sum_{i=1}^l a_i k(\cdot, \mathbf{x}_i) \right\|_{\mathcal{H}}^2 + \epsilon \|\mathbf{a}\|_1 \rightarrow \min_{\mathbf{a} \in \mathbb{R}^l}, \quad (2.44)$$

where $\epsilon > 0$. (2.44) is an extension of the Lasso problem [4]: the second $\|\mathbf{a}\|_1$ induces sparsity. However, as opposed to the standard Lasso cost the first term measures the approximation error on training sample making use of the $\|\cdot\|_{\mathcal{H}}^2$ RKHS norm and not the standard Euclidean one. Let us further assume that $\langle f, 1 \rangle_{\mathcal{H}} = 0^1$ and for the tradeoff parameter of SVM, $C \rightarrow \infty$. Let us decompose the searched coefficient \mathbf{a} into its positive and negativ part, i.e.,

$$\mathbf{a} = \mathbf{a}^+ - \mathbf{a}^-, \quad (2.45)$$

where $\mathbf{a}^+, \mathbf{a}^- \geq \mathbf{0}$ and $\mathbf{a}^+ \circ \mathbf{a}^- = \mathbf{0}$. [83] proved that in this case, the (2.44) and (2.43) problems are equivalent, in the sense, that the solution of (2.44), the $(\mathbf{a}^+, \mathbf{a}^-)$ pair is identical to that of $(\mathbf{d}^*, \mathbf{d})$, the optimal solution of the dual SVM problem. The equivalence of sparse coding and SVMs can also be extended to the noisy case by considering a larger RKHS space encapsulating the noise process [84].

Both works [83, 84] however transform the insensitivity parameter (ϵ) into a ‘uniform’ sparsification, that is into the weight of the sparsity-inducing regularization term (compare, e.g., (2.44) and (2.43)). Our question was, whether it is possible to transform the insensitivity ϵ into *component-wise* sparsity-inducing regularization. To address this problem, we first define the extended (\mathbf{c}, \mathbf{e}) -SVM and (\mathbf{p}, \mathbf{s}) -sparse tasks, then the correspondence of these two problems enabling component-wise ϵ -sparsity inducing is derived.

The (\mathbf{c}, \mathbf{e}) -SVM Task

Below, we introduce an extended SVM problem family. For notational simplicity, instead of approximating in semi-parametric form (e.g., $g + b$, where $g \in \mathcal{H}$), we shall deal with the so-called non-parametric scheme ($g \in \mathcal{H}$). This approach is also well grounded by the representer theorem of kernel based approximations [148].

¹ This restriction gives rise to constraint $\sum_i a_i = 0$.

The usual SVM task, (2.41) is modified as follows:

1. We approximate in the form $f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}}$.
2. We shall use approximation errors and weights that may differ for each sample point.

Introducing vector \mathbf{e} for the ϵ -insensitive costs and \mathbf{c} for the weights, respectively, the generalized problem is defined as:

$$\sum_{i=1}^l c_i |y_i - f_{\mathbf{w}}(\mathbf{x}_i)|_{e_i} + \frac{1}{2} \|\mathbf{w}\|_{\mathcal{H}}^2 \rightarrow \min_{\mathbf{w} \in \mathcal{H}}, \quad (\mathbf{c} > \mathbf{0}, \mathbf{e} \geq \mathbf{0}). \quad (2.46)$$

This problem is referred to as the (\mathbf{c}, \mathbf{e}) -SVM task. The original task of Eq. (2.41) corresponds to the particular choice of $(C\mathbf{1}, \epsilon\mathbf{1})$ and $b = 0$. Alike to the original SVM problem, the (\mathbf{c}, \mathbf{e}) -SVM task also has its quadratic equivalent in the dual space, which is as follows

$$\begin{aligned} \frac{1}{2} (\mathbf{d}^* - \mathbf{d})^T \mathbf{G} (\mathbf{d}^* - \mathbf{d}) - (\mathbf{d}^* - \mathbf{d})^T \mathbf{y} + (\mathbf{d}^* + \mathbf{d})^T \mathbf{e} &\rightarrow \min_{\mathbf{d}^* \in \mathbb{R}^l, \mathbf{d} \in \mathbb{R}^l}, \\ \text{subject to } \{ \mathbf{c} \geq \mathbf{d}^*, \mathbf{d} \geq \mathbf{0} \}, \end{aligned} \quad (2.47)$$

where \mathbf{G} denotes the Gram matrix of kernel k on the $\{\mathbf{x}_i\}_{i=1}^l$ sample points. Moreover, the optimal \mathbf{w} and the $f_{\mathbf{w}}(\mathbf{x})$ regression function can be expressed making use of the obtained $(\mathbf{d}, \mathbf{d}^*)$ dual solution as

$$\mathbf{w} = \sum_{i=1}^l (d_i - d_i^*) \boldsymbol{\varphi}(\mathbf{x}_i), \quad (2.48)$$

$$f_{\mathbf{w}}(\mathbf{x}) = \left\langle \sum_{i=1}^l (d_i - d_i^*) \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}) \right\rangle_{\mathcal{H}} = \sum_{i=1}^l (d_i - d_i^*) k(\mathbf{x}, \mathbf{x}_i). \quad (2.49)$$

Let us notice that the optimal solution $f_{\mathbf{w}}(\cdot)$ can be expressed as the linear combination of $k(\cdot, \mathbf{x}_i)$ s. This is the form that is guaranteed by the representer theorem [148] under mild conditions on the cost function—the coefficient are of course problem always problem specific.

The (\mathbf{p}, \mathbf{s}) -Sparse Task

Below, we introduce an extended sparse coding scheme in RKHSs. Indeed, let us consider the optimization problem

$$F(\mathbf{a}) = \frac{1}{2} \left\| f(\cdot) - \sum_{i=1}^l a_i k(\cdot, \mathbf{x}_i) \right\|_{\mathcal{H}}^2 + \sum_{i=1}^l p_i |a_i|_{s_i} \rightarrow \min_{\mathbf{a} \in \mathbb{R}^l}, \quad (\mathbf{p} > 0, \mathbf{s} \geq 0) \quad (2.50)$$

whose goal is to approximate objective function $f \in \mathcal{H} = \mathcal{H}(k)$ on the sample points $\{\mathbf{x}_i, y_i\}_{i=1}^l$. This problem is referred to as the \mathbf{p} -weighted and \mathbf{s} -sparse task, or the (\mathbf{p}, \mathbf{s}) -sparse task, for short. For the particular choice of $(\epsilon \mathbf{1}, \mathbf{0})$ we get back the sparse representation form of Eq. (2.44).

Correspondence of the (\mathbf{c}, \mathbf{e}) -SVM and (\mathbf{p}, \mathbf{s}) -Sparse Problems

One can derive a correspondence between the (\mathbf{c}, \mathbf{e}) -SVM and (\mathbf{p}, \mathbf{s}) -sparse problems. Our result [149], which achieves component-wise ϵ -sparsity inducing, is summarized in the following proposition:

Proposition 1. *Let \mathcal{X} denote a non-empty set, let k be a reproducing kernel on \mathcal{X} , and let us given samples $\{\mathbf{x}_i, y_i\}_{i=1}^l$, where $\mathbf{x}_i \in \mathcal{X}, y_i \in \mathbb{R}$. Assume further that the values of the RKHS target function $f \in \mathcal{H} = \mathcal{H}(k)$ can be observed in points \mathbf{x}_i ($f(\mathbf{x}_i) = y_i$) and let $f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{x}) \rangle_{\mathcal{H}}$. Then the duals of the (\mathbf{c}, \mathbf{e}) -SVM task [(2.46)] and that of the (\mathbf{p}, \mathbf{s}) -sparse task [(2.50)] can be transformed into each other by the generalized inverse \mathbf{G}^- of the Gram matrix $\mathbf{G} = [G_{i,j}]_{i,j=1}^l = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^l$ via $(\mathbf{d}^*, \mathbf{d}, \mathbf{G}, \mathbf{y}) \leftrightarrow (\mathbf{d}^+, \mathbf{d}^-, \mathbf{G}^- \mathbf{G} \mathbf{G}^-, \mathbf{G}^- \mathbf{y}) = (\mathbf{d}^+, \mathbf{d}^-, \mathbf{G}^-, \mathbf{G}^- \mathbf{y})$. [For proof, see Appendix A.2.]*

2.3 Multilayer Kerceptron

Now, we embed support vector machines to multilayer perceptrons. In Section 2.3.1 we briefly introduce multilayer perceptrons (MLP). We present our novel multilayer kerceptron architecture in Section 2.3.2. In Section 2.3.3, we extend the backpropagation method of MLPs to multilayer kerceptrons.

2.3.1 Multilayer Perceptron

The multilayer perceptron (MLP) network [88] is a multilayer approximating scheme, where each layer of the network performs the nonlinear mapping

$$\mathbf{x} \mapsto \mathbf{g}(\mathbf{W}\mathbf{x}). \quad (2.51)$$

These ‘simple’ mappings are the composition of linear transformation \mathbf{W} , followed by the differentiable, nonlinear mapping \mathbf{g} . Typical choice for \mathbf{g} is a coordinate-wise acting sigmoid function. In the MLP task, the goal is to tune matrices \mathbf{W} of the network to approximate the sampled input-output mapping given by input-output training pairs $\{\mathbf{x}(t), \mathbf{d}(t)\}$, where $\mathbf{x}(t) \in \mathcal{X} = \mathbb{R}^{d_1}$, $\mathbf{d}(t) \in \mathbb{R}^{d_2}$. In an adaptive approach, the MLP task is to continuously minimize the instantaneous squared error function

$$\varepsilon^2(t) = \|\mathbf{d}(t) - \mathbf{y}(t)\|_2^2 \rightarrow \min_{\mathbf{W}_1, \dots, \mathbf{W}_L}, \quad (2.52)$$

where $\mathbf{y}(t) \in \mathbb{R}^{d_2}$ denotes the output of the network at time t , the estimation for $\mathbf{d}(t)$. The optimization of (2.52) can be carried out by, e.g., making use of the stochastic gradient descent technique. In the resulting optimization, the errors for a given layer (\mathbf{W}_l) are propagated back from the subsequent layer (\mathbf{W}_{l+1}), this is the well-known backpropagation algorithm.

2.3.2 The Multilayer Kerceptron Architecture

Now, we embed support vector machines to MLPs. To do so, first let us notice that the mapping of a general MLP layer [(2.51)] can be written as

$$\mathbf{x} \mapsto \mathbf{g} \left(\begin{bmatrix} \vdots \\ \langle \mathbf{w}_i, \mathbf{x} \rangle \\ \vdots \end{bmatrix} \right), \quad (2.53)$$

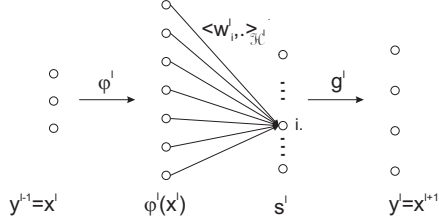


Figure 2.1: The l^{th} layer of the MLK, $l = 1, 2, \dots, L$. The input (\mathbf{x}^l) of each layer is the output of the preceding layer (\mathbf{y}^{l-1}). The external world is the 0^{th} layer providing input to the first layer of the MLK. Inputs \mathbf{x}^l to layer l are mapped by features mapping φ^l undergo scalar product by the weights (\mathbf{w}_i^l) of the layer in RKHS $\mathcal{H}^l = \mathcal{H}^l(k^l)$. The result is vector \mathbf{s}^l , which undergoes nonlinear processing \mathbf{g}^l , with a differentiable function. The output of this nonlinear function is the input to the next layer, layer \mathbf{x}^{l+1} . The output of the network is the output of the last layer.

where \mathbf{w}_i^T denotes the i^{th} row of matrix \mathbf{W} . Let us now replace the scalar product terms $\langle \mathbf{w}_i, \mathbf{x} \rangle$ with $\langle \mathbf{w}_i, \varphi(\mathbf{x}) \rangle_{\mathcal{H}}$ and define the general layer of the network as²

$$\mathbf{x} \mapsto \mathbf{g} \left(\begin{bmatrix} \langle \mathbf{w}_1, \varphi(\mathbf{x}) \rangle_{\mathcal{H}} \\ \vdots \\ \langle \mathbf{w}_N, \varphi(\mathbf{x}) \rangle_{\mathcal{H}} \end{bmatrix} \right). \quad (2.54)$$

A network made of such layers will be called multilayer kerception (MLK). For an illustration of the MLK network, see Fig. 2.1. In MLK, the input (\mathbf{x}^l) of each layer is the output of the preceding layer (\mathbf{y}^{l-1}). The external world is the 0^{th} layer providing input to the first layer of the MLK. $\mathbf{x}^l = \mathbf{y}^{l-1} \in \mathbb{R}^{N_l^i}$, where N_l^i is the input dimension of the l^{th} layer. Inputs \mathbf{x}^l to layer l are mapped by features φ^l and are multiplied by the weights \mathbf{w}_i^l . This two-step process can be accomplished implicitly by making use of kernel k^l and the expansion property for \mathbf{w}_i^l s. The result is vector $\mathbf{s}^l \in \mathbb{R}^{N_l^s}$, which undergoes nonlinear processing \mathbf{g}^l , where function \mathbf{g}^l is differentiable. The output of this nonlinear function is the input to the next layer, i.e., layer \mathbf{x}^{l+1} . The output of the last layer (layer L , the output of the network) will be referred to as \mathbf{y} . Given that $\mathbf{y}^l = \mathbf{x}^{l+1} \in \mathbb{R}^{N_o^l}$, the output dimension of layer l is

² We assume that the sample space \mathcal{X} is the finite dimensional Euclidean space.

N_o^l .

2.3.3 Backpropagation of Multilayer Kerceptrons

Below, we show that (i) the backpropagation method of MLPs can be extended to MLKs and it (ii) be accomplished in dual space requiring kernel computations only.

We consider a slightly more general task, which incorporates regularizing terms:

$$c(t) = \varepsilon^2(t) + r(t) \longrightarrow \min_{\{\mathcal{H}^{cl} \ni \mathbf{w}_i^l; l=1,\dots,L; i=1,\dots,N_S^l\}}, \quad (2.55)$$

where

$$\varepsilon^2(t) = \|\mathbf{d}(t) - \mathbf{y}(t)\|_2^2, \quad (2.56)$$

$$r(t) = \sum_{l=1}^L \sum_{i=1}^{N_S^l} \lambda_i^l \|\mathbf{w}_i^l(t)\|_{\mathcal{H}^{cl}}^2 \quad (\lambda_i^l \geq 0) \quad (2.57)$$

are the approximation and the regularization terms of the cost function, respectively, and $\mathbf{y}(t)$ denotes the output of the network for the t^{th} input. Parameters λ_i^l control the trade-off between approximation and regularization. For $\lambda_i^l = 0$ the best approximation is searched like in the MLP task [(2.52)]. With these notations at hand, we can present our results [150] now.

Proposition 2 (explicit case). *Let us suppose that the $\mathbf{x} \mapsto \langle \mathbf{w}, \boldsymbol{\varphi}^l(\mathbf{x}) \rangle_{\mathcal{H}^{cl}}$ and the \mathbf{g}^l functions are differentiable ($l = 1, \dots, L$). Then, backpropagation rule can be derived for MLK with cost function (2.55).*

Proposition 3 (implicit case). *Assume that the following holds*

1. *Constraint on differentiability: Kernels k^l are differentiable with respect to both arguments and functions \mathbf{g}^l are also differentiable ($l = 1, \dots, L$).*
2. *Expansion property: The initial weights $\mathbf{w}_i^l(1)$ of the network can be ex-*

pressed in the dual representation, i.e.,

$$\mathcal{H}^l \ni \mathbf{w}_i^l(1) = \sum_{j=1}^{N_i^l(1)} \alpha_{i,j}^l(1) \varphi^l(\mathbf{z}_{i,j}^l(1)) \quad (l = 1, \dots, L; i = 1, \dots, N_S^l). \quad (2.58)$$

Then backpropagation can be derived for MLK with cost function (2.55). This procedure preserves the expansion property (2.58), which then remains valid for the tuned network. The algorithm is implicit in the sense that it can be realized in the dual space, using kernel computations only.

The pseudo codes of the MLK backpropagation algorithms are provided in Table 2.5 and Table 2.6, respectively. The MLK backpropagation can be envisioned as follows (see Table 2.5 and 2.6 simultaneously):

1. backpropagated error $\delta^l(t)$ starts from $\delta^L(t)$ and is computed by a backward recursion via the differential expression $\frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]}$.
2. expression $\frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]}$ can be determined by means of feature mapping φ^{l+1} , or, in an implicit fashion, through kernels k^{l+1} .
3. two components play roles in the tuning of \mathbf{w} -s:
 - (a) *forgetting* is accomplished by scaling the weights \mathbf{w}_i^l with multiplier $1 - 2\mu_i^l(t)\lambda_i^l$, where λ_i^l is the regularization coefficient.
 - (b) *adaptation* occurs through the backpropagated error. Weights at layer l are tuned by feature space representation of $\mathbf{x}^l(t)$, the actual input arriving at layer l . Tuning is weighted by the backpropagated error.

Derivations of these algorithms are provided in Appendix A.3.

Table 2.5: Pseudocode of the explicit MLK backpropagation algorithm.

Inputs

sample points: $\{\mathbf{x}(t), \mathbf{d}(t)\}_{t=1, \dots, T, T}$
 cost function: $\lambda_i^l \geq 0$ ($l = 1, \dots, L; i = 1, \dots, N_S^l$)
 learning rates: $\mu_i^l(t) > 0$ ($l = 1, \dots, L; i = 1, \dots, N_S^l; t = 1, \dots, T$)

Network initialization

size: L (number of layers), N_1^l, N_S^l, N_o^l ($l = 1, \dots, L$)
 parameters: $\mathbf{w}_i^l(1)$ ($l = 1, \dots, L; i = 1, \dots, N_S^l$)

Start computation

Choose sample $\mathbf{x}(t)$

Feedforward computation

$\mathbf{x}^l(t)$ ($l = 2, \dots, L+1$), $\mathbf{s}^l(t)$ ($l = 2, \dots, L$)^a

Backpropagation of error

$l = L$

while $l \geq 1$

if ($l = L$)

$$\boldsymbol{\delta}^L(t) = 2 [\mathbf{y}(t) - \mathbf{d}(t)]^T (\mathbf{g}^L)' (\mathbf{s}^L(t))$$

else

$$\frac{\partial [\mathbf{s}^{l+1}(t)]}{\partial [\mathbf{s}^l(t)]} = \begin{bmatrix} \vdots \\ \frac{\partial [\langle \mathbf{w}_i^{l+1}(t), \boldsymbol{\varphi}^{l+1}(\mathbf{u}) \rangle]_{\beta(l+1)}}{\partial [\mathbf{u}]} \Big|_{\mathbf{u}=\mathbf{x}^{l+1}(t)} \\ \vdots \end{bmatrix} (\mathbf{g}^l)' (\mathbf{s}^l(t))^b$$

$$\boldsymbol{\delta}^l(t) = \boldsymbol{\delta}^{l+1}(t) \frac{\partial [\mathbf{s}^{l+1}(t)]}{\partial [\mathbf{s}^l(t)]}$$

end

Weight update

for all i : $1 \leq i \leq N_S^l$

$$\mathbf{w}_i^l(t+1) = (1 - 2\mu_i^l(t)\lambda_i^l)\mathbf{w}_i^l(t) - \mu_i^l(t)\delta_i^l(t)\boldsymbol{\varphi}^l(\mathbf{x}^l(t))$$

$l = l - 1$

End computation

^a The output of the network, i.e., $\mathbf{y}(t) = \mathbf{x}^{L+1}(t)$ is also computed.

^b Here: $i = 1, \dots, N_S^{l+1}$.

Table 2.6: Pseudocode of the implicit MLK backpropagation algorithm.

<p>Inputs</p> <p>sample points: $\{\mathbf{x}(t), \mathbf{d}(t)\}_{t=1,\dots,T,T}$</p> <p>cost function: $\lambda_i^l \geq 0$ ($l = 1, \dots, L; i = 1, \dots, N_S^l$)</p> <p>learning rates: $\mu_i^l(t) > 0$ ($l = 1, \dots, L; i = 1, \dots, N_S^l; t = 1, \dots, T$)</p> <p>Network initialization</p> <p>size: L (number of layers), N_I^l, N_S^l, N_O^l ($l = 1, \dots, L$)</p> <p>parameters: $\mathbf{w}_i^l(1)$-expansions ($l = 1, \dots, L; i = 1, \dots, N_S^l$)</p> <p>coefficients: $\alpha_i^l(1) \in \mathbb{R}^{N_I^l(1)}$</p> <p>ancestors: $\mathbf{z}_{i,j}^l(1)$, where $j = 1, \dots, N_I^l(1)$</p> <p>Start computation</p> <p>Choose sample $\mathbf{x}(t)$</p> <p>Feedforward computation</p> <p>$\mathbf{x}'(t)$ ($l = 2, \dots, L+1$), $\mathbf{s}^l(t)$ ($l = 2, \dots, L$)^a</p> <p>Backpropagation of error</p> <p>$l = L$</p> <p>while $l \geq 1$</p> <p> if ($l = L$)</p> <p> $\delta^L(t) = 2 [\mathbf{y}(t) - \mathbf{d}(t)]^T (\mathbf{g}^L)' (\mathbf{s}^L(t))$</p> <p> else</p> <p> $\frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]} = \begin{bmatrix} \vdots \\ \sum_{j=1}^{N_I^{l+1}(t)} \alpha_{ij}^{l+1}(t) [k^{l+1}]'_{iy} (\mathbf{z}_{ij}^{l+1}(t), \mathbf{x}^{l+1}(t)) \\ \vdots \end{bmatrix} (\mathbf{g}^l)' (\mathbf{s}^l(t))^b$</p> <p> $\delta^l(t) = \delta^{l+1}(t) \frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]}$</p> <p> end</p> <p>Weight update</p> <p> for all $i: 1 \leq i \leq N_S^l$</p> <p> $N_i^l(t+1) = N_i^l(t) + 1$</p> <p> $\alpha_i^l(t+1) = [(1 - 2\mu_i^l(t)\lambda_i^l) \alpha_i^l(t); -\mu_i^l(t)\delta_i^l(t)]$</p> <p> $\mathbf{z}_{i,j}^l(t+1) = \mathbf{z}_{i,j}^l(t)$ ($j = 1, \dots, N_I^l(t)$)</p> <p> $\mathbf{z}_{i,j}^l(t+1) = \mathbf{x}^l(t)$ ($j = N_I^l(t+1)$)</p> <p> $l = l - 1$</p> <p>End computation</p>	
--	--

^a The output of the network, i.e., $\mathbf{y}(t) = \mathbf{x}^{L+1}(t)$ is also computed.

^b $i = 1, \dots, N_S^{l+1}$. Note also that $(k^l)'_{iy}$ denotes the derivative of kernel k^l according to its second argument.

Chapter 3

Theory – Independent Subspace Based Dictionary Learning

In this chapter we present our novel independent subspace based dictionary learning approaches. Contrary to Chapter 2, where the underlying assumption for the hidden sources was *sparsity and structured sparsity*, here we are dealing with *independent* non-Gaussian sources. In Section 3.1 we unify controlled dynamical systems and independent subspace based dictionary learning. Section 3.2 is about the extension of the current ISA models to the partially observable case. In Section 3.3 and Section 3.4 we are dealing with complex and nonparametric generalizations, respectively. Section 3.5 is devoted to the convolutive case. We note that the different methods can be used in combinations, too. For all the introduced models, we derive separation principle based solution. These separation principles make it possible to estimate the models even in case of different, or unknown dimensional independent source components. In Section 3.6 we present a novel random projection based, parallel estimation technique for high dimensional information theoretical quantities. Numerical experiments demonstrating the efficiency of our methods are given in Chapter 5.

3.1 Controlled Models

The traditional ICA/ISA problem family can model *hidden* independent variables, but does not allow/handle *control* variables. In this section we couple ISA based dictionary learning methods with control variables. To emphasize the fact that we are dealing with sources having dynamics, in the sequel, we will refer to such problems as independent process analysis (IPA)—instead of ISA.

In our approach will adapt the D-optimal identification of ARX (autoregressive with exogenous input) dynamical systems, that we briefly summarize in Section 3.1.1. Section 3.1.2 defines the problem domain, the ARX-IPA task. Our solution technique for the ARX-IPA problem is derived in Section 3.1.3.

3.1.1 D-optimal Identification of ARX Models

We sketch the basic thoughts that lead to D-optimal identification of ARX models. The dynamical system to be identified is fully observed and evolves according to the ARX equation

$$\mathbf{s}_{t+1} = \sum_{i=0}^{L_s-1} \mathbf{F}_i \mathbf{s}_{t-i} + \sum_{j=0}^{L_u-1} \mathbf{B}_j \mathbf{u}_{t+1-j} + \mathbf{e}_{t+1}, \quad (3.1)$$

where (i) $\mathbf{s} \in \mathbb{R}^{D_s}$, $\mathbf{e} \in \mathbb{R}^{D_e}$ ($D_s = D_e$) represent the state of the system and the noise, respectively, (ii) $\mathbf{u} \in \mathbb{R}^{D_u}$ represents the control variables, and (iii) polynomial matrix (given by matrices $\mathbf{F}_i \in \mathbb{R}^{D_s \times D_s}$ and identity matrix \mathbf{I})

$$\mathbf{F}[z] = \mathbf{I} - \sum_{i=0}^{L_s-1} \mathbf{F}_i z^{i+1} \in \mathbb{R}[z]^{D_s \times D_s} \quad (3.2)$$

is stable, that is

$$\det(\mathbf{F}[z]) \neq 0, \quad (3.3)$$

for all $z \in \mathbb{C}, |z| \leq 1$. Our task is (i) the efficient estimation of parameters $\Theta = [\mathbf{F}_0, \dots, \mathbf{F}_{L_s-1}, \mathbf{B}_0, \dots, \mathbf{B}_{L_u-1}]$ that determine the dynamics and (ii) noise \mathbf{e} that drives the process by the ‘optimal choice’ of control values \mathbf{u} . Formally, D-optimality

aims to maximize one of the two objectives

$$J_{par}(\mathbf{u}_{t+1}) = I(\boldsymbol{\Theta}, \mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{s}_{t-1}, \dots, \mathbf{u}_{t+1}, \mathbf{u}_t, \dots), \quad (3.4)$$

$$J_{noise}(\mathbf{u}_{t+1}) = I(\mathbf{e}_{t+1}, \mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{s}_{t-1}, \dots, \mathbf{u}_{t+1}, \mathbf{u}_t, \dots) \quad (3.5)$$

for $\mathbf{u}_{t+1} \in U$. In other words, we choose control value \mathbf{u} from the achievable domain U (e.g., from a box domain) such that it maximizes the mutual information between the next observation and the parameters (or the driving noise) of the system. It can be shown [151], that if (i) $\boldsymbol{\Theta}$ has matrix Gaussian, (ii) \mathbf{e} has Gaussian, and the covariance matrix of \mathbf{e} has inverted Wishart distributions, then in the Bayesian setting, maximization of the J objectives can be reduced to the solution of a quadratic programming task, priors of $\boldsymbol{\Theta}$ and \mathbf{e} remain in their supposed distribution family and undergo simple updating. The considerations allow for control, but assume full observability about the state variables. Now, we extend the method to hidden variables in the ARX-IPA model of the next section.

3.1.2 The ARX-IPA Problem

In the ARX-IPA model we assume that state \mathbf{s} of the system cannot be observed directly, but its linear and unknown mixture (\mathbf{x}) is available for observation [152]:

$$\mathbf{s}_{t+1} = \sum_{i=0}^{L_s-1} \mathbf{F}_i \mathbf{s}_{t-i} + \sum_{j=0}^{L_u-1} \mathbf{B}_j \mathbf{u}_{t+1-j} + \mathbf{e}_{t+1}, \quad (3.6)$$

$$\mathbf{x}_t = \mathbf{A} \mathbf{s}_t, \quad (3.7)$$

where L_s and L_u denote the number of the $\mathbf{F}_i \in \mathbb{R}^{D_s \times D_s}$, $\mathbf{B}_j \in \mathbb{R}^{D_s \times D_u}$ matrices in the corresponding sums. We assume

- for the $\mathbf{e}^m \in \mathbb{R}^{d_m}$ components of $\mathbf{e} = [\mathbf{e}^1; \dots; \mathbf{e}^M] \in \mathbb{R}^{D_s}$ ($D_s = \sum_{m=1}^M d_m$) that at most one of them may be Gaussian, their temporal evolution is i.i.d. (independent identically distributed), and $I(\mathbf{e}^1; \dots; \mathbf{e}^M) = 0$, that is, they satisfy the *ISA assumptions*.¹

¹By d_m -dimensional \mathbf{e}^m components, we mean that \mathbf{e}^m s cannot be decomposed into smaller dimensional independent parts. This property is called *irreducibility* in [153].

- that the polynomial matrix $\mathbf{F}[z] = \mathbf{I} - \sum_{i=0}^{L_s-1} \mathbf{F}_i z^{i+1}$ is stable and the mixing matrix $\mathbf{A} \in \mathbb{R}^{D_s \times D_s}$ is invertible. We note, that compared to Chapter 2, in the presented ISA based models the mixing matrix \mathbf{A} plays the role of the dictionary.

The ARX-IPA task is to estimate the unknown mixing matrix \mathbf{A} , parameters $\{\mathbf{F}_i\}_{i=0}^{L_s-1}$, $\{\mathbf{B}_j\}_{j=0}^{L_u-1}$, \mathbf{s} and \mathbf{e} by means of observations \mathbf{x} only.

In the special case of $L_s = L_u = 0$, that is

$$\mathbf{x} = \mathbf{A}\mathbf{e} \quad (3.8)$$

we get back the traditional ISA problem, where the goal is estimate the mixing matrix \mathbf{A} and the hidden source \mathbf{e} , and there is no control. If $d_m = 1$ ($\forall m$) also holds in ISA, i.e., the independent \mathbf{e}^m source components are one-dimensional, we obtain the ICA problem.

3.1.3 Identification Method for ARX-IPA

Below, we solve the ARX-IPA model, i.e., we include the control variables in IPA. We derive a separation principle based solution by transforming the estimation into two subproblems: to that of a fully observed model (Section 3.1.1) and an ISA task.

One can apply the basis transformation rule of ARX processes and use (3.6) and (3.7) repeatedly to get

$$\mathbf{x}_{t+1} = \sum_{i=0}^{L_s-1} (\mathbf{A}\mathbf{F}_i\mathbf{A}^{-1})\mathbf{x}_{t-i} + \sum_{j=0}^{L_u-1} (\mathbf{A}\mathbf{B}_j)\mathbf{u}_{t+1-j} + (\mathbf{A}\mathbf{e}_{t+1}). \quad (3.9)$$

According to the d-dependent central limit theorem [154] the marginals of $\mathbf{A}\mathbf{e}_{t+1}$ are approximately Gaussian and thus the parameters $(\{\mathbf{A}\mathbf{F}_i\mathbf{A}^{-1}\}_{i=0}^{L_s-1}, \{\mathbf{A}\mathbf{B}_j\}_{j=0}^{L_u-1})$ and the noise $(\mathbf{A}\mathbf{e}_{t+1})$ of process \mathbf{x} can be estimated by means of the D-optimality principle that assumes a fully observed process. The estimation of $\mathbf{A}\mathbf{e}_{t+1}$ can be seen as the observation of an ISA problem because components \mathbf{e}^m of \mathbf{e} are independent. ISA techniques can be used to identify \mathbf{A} and then from the estimated parameters of process \mathbf{x} , the estimations of \mathbf{F}_i and \mathbf{B}_j follow.

Note:

1. In the above described ARX-IPA technique, the D-optimal ARX procedure is an *online* estimation for the innovation $\varepsilon = \mathbf{A}\mathbf{e}$, the input of the ISA method. To the best of our knowledge, there is no existing online ISA method in the literature. However, having such a procedure, one can easily integrate it into the presented approach to get a fully online ARX-IPA estimation scheme.
2. Similar ideas can be used for the estimation of an ARMAX-IPA [155], or post nonlinear model [156]. In the ARMAX-IPA model, the state equation (3.6) is generalized to $L_e \geq 0$, i.e.,

$$\mathbf{s}_{t+1} = \sum_{i=0}^{L_s-1} \mathbf{F}_i \mathbf{s}_{t-i} + \sum_{j=0}^{L_u-1} \mathbf{B}_j \mathbf{u}_{t+1-j} + \mathbf{e}_{t+1} + \sum_{k=0}^{L_e-1} \mathbf{H}_k \mathbf{e}_{t-k}. \quad (3.10)$$

In this case, we assume additionally that the polynomial matrix $\mathbf{H}[z] = \mathbf{I} + \sum_{k=1}^{L_e} \mathbf{H}_k z^k \in \mathbb{R}[z]^{D_s \times D_s}$ is stable.² In the PNL ARX-IPA model, the observation equation (3.7) is generalized to

$$\mathbf{x}_t = \mathbf{f}(\mathbf{A}\mathbf{s}_t), \quad (3.11)$$

where \mathbf{f} is an unknown, but component-wise acting invertible mapping.

3.2 Incompletely Observable Models

The goal of this section is to search for independent multidimensional processes subject to missing and mixed observations. In spite of the popularity of ICA and its numerous successful applications, the case of missing observation has been considered only for the simplest ICA model in the literature [133, 134]. In this section we extend the solution to (i) multidimensional sources (ISA) and (ii) ease the i.i.d. constraint; we consider AR independent process analysis (AR-IPA problem).

²Note that this requirement is automatically fulfilled for $L_e = 0$, when $\mathbf{H}[z] = \mathbf{I}$.

3.2.1 The AR-IPA Model with Missing Observations

We define the AR-IPA model for missing observations (mAR-IPA) [157, 158]. Let us assume that we can only partially (at certain coordinates/time instants) observe (\mathbf{y}) the mixture (\mathbf{x}) of independent AR sources, that is

$$\mathbf{s}_{t+1} = \sum_{l=0}^{L_s-1} \mathbf{F}_l \mathbf{s}_{t-l} + \mathbf{e}_{t+1}, \quad \mathbf{x}_t = \mathbf{A} \mathbf{s}_t, \quad \mathbf{y}_t = \mathcal{M}_t(\mathbf{x}_t), \quad (3.12)$$

where

- the driving noises, or the innovations $\mathbf{e}^m \in \mathbb{R}^{d_m}$ ($\mathbf{e} = [\mathbf{e}_1; \dots; \mathbf{e}_M] \in \mathbb{R}^D$) of the hidden source $\mathbf{s} \in \mathbb{R}^D$ ($D = \sum_{m=1}^M d_m$) are independent, at least one of them is Gaussian, and i.i.d. in time, i.e., they satisfy the ISA assumptions.
- the unknown mixing matrix $\mathbf{A} \in \mathbb{R}^{D \times D}$ is invertible,
- the AR dynamics $\mathbf{F}[z] = \mathbf{I} - \sum_{l=0}^{L_s-1} \mathbf{F}_l z^{l+1} \in \mathbb{R}[z]^{D \times D}$ is stable and
- the \mathcal{M}_t ‘mask mappings’ represent the coordinates and the time indices of the non-missing observations.

Our task is the estimation of the hidden source \mathbf{s} and the mixing matrix \mathbf{A} (or its inverse \mathbf{W}) from observation \mathbf{y} . For the special choice of $\mathcal{M}_t = \text{identity}$ ($\forall t$), the AR-IPA problem [159] is obtained. If $L_s = 0$ also hold, we get the ISA task.

3.2.2 Identification Method for mAR-IPA

The mAR-IPA identification can be accomplished as follows. The \mathbf{x}_t is an invertible linear transformation of the hidden AR process \mathbf{s} and thus \mathbf{x}_t is also an AR process with innovation $\mathbf{A} \mathbf{e}_{t+1}$:

$$\mathbf{x}_{t+1} = \sum_{l=0}^{L_s-1} \mathbf{A} \mathbf{F}_l \mathbf{A}^{-1} \mathbf{x}_{t-l} + \mathbf{A} \mathbf{e}_{t+1}. \quad (3.13)$$

According to the d-dependent central limit theorem [154], the marginals of variable $\mathbf{A} \mathbf{e}$ are approximately Gaussian, so one carry out the estimation by

1. identifying the partially observed AR process \mathbf{y}_t , and then by
2. estimating the independent components \mathbf{e}^m from the estimated innovation by means of ISA.

3.3 Complex Models

Current methods in the ICA literature are only capable of coping with one-dimensional complex independent sources, i.e., with the simplest ICA model. In this section by extending the independent subspace analysis model to complex variables, we make it possible to tackle problems with multidimensional independent sources. First we summarize a few basic concepts for complex random variables (Section 3.3.1). In Section 3.3.2 the complex ISA model is introduced. In Section 3.3.3 we show, that under certain non-Gaussian assumptions the solution of the complex ISA problem can be reduced to the solution of a real ISA problem.

3.3.1 Complex Random Variables

Below we summarize a few basic concepts of complex random variables, define two mappings that will be useful in the next section and note that an excellent review on this topic can be found in [160].

A complex random variable $\mathbf{v} \in \mathbb{C}^L$ is defined as a random variable of the form $\mathbf{v} = \mathbf{v}_R + i\mathbf{v}_I \in \mathbb{C}^L$, where the real and imaginary parts of \mathbf{v} , i.e., $\mathbf{v}_R \in \mathbb{R}^L$ and $\mathbf{v}_I \in \mathbb{R}^L$ are real vector random variables. Let us define the $\varphi_v : \mathbb{C}^L \mapsto \mathbb{R}^{2L}$, $\varphi_M : \mathbb{C}^{L_1 \times L_2} \mapsto \mathbb{R}^{2L_1 \times 2L_2}$ mappings as

$$\varphi_v(\mathbf{v}) = \mathbf{v} \otimes \begin{bmatrix} \Re(\cdot) \\ \Im(\cdot) \end{bmatrix}, \quad \varphi_M(\mathbf{M}) = \mathbf{M} \otimes \begin{bmatrix} \Re(\cdot) & -\Im(\cdot) \\ \Im(\cdot) & \Re(\cdot) \end{bmatrix}, \quad (3.14)$$

where \Re stands for the real part, \Im for the imaginary part, subscript ' v ' (' M ') for vector (matrix) and \otimes is the Kronecker product. Known properties of mappings φ_v ,

φ_M are as follows [161]:

$$\det[\varphi_M(\mathbf{M})] = |\det(\mathbf{M})|^2 \quad (\mathbf{M} \in \mathbb{C}^{L \times L}), \quad (3.15)$$

$$\varphi_M(\mathbf{M}_1 \mathbf{M}_2) = \varphi_M(\mathbf{M}_1) \varphi_M(\mathbf{M}_2) \quad (\mathbf{M}_1 \in \mathbb{C}^{L_1 \times L_2}, \mathbf{M}_2 \in \mathbb{C}^{L_2 \times L_3}), \quad (3.16)$$

$$\varphi_v(\mathbf{M} \mathbf{v}) = \varphi_M(\mathbf{M}) \varphi_v(\mathbf{v}) \quad (\mathbf{M} \in \mathbb{C}^{L_1 \times L_2}, \mathbf{v} \in \mathbb{C}^{L_2}), \quad (3.17)$$

$$\varphi_M(\mathbf{M}_1 + \mathbf{M}_2) = \varphi_M(\mathbf{M}_1) + \varphi_M(\mathbf{M}_2) \quad (\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{C}^{L_1 \times L_2}), \quad (3.18)$$

$$\varphi_M(c\mathbf{M}) = c\varphi_M(\mathbf{M}) \quad (\mathbf{M} \in \mathbb{C}^{L_1 \times L_2}, c \in \mathbb{R}). \quad (3.19)$$

In words: (3.15) describes transformation of determinant, while (3.16), (3.17), (3.18) and (3.19) expresses preservation of operation for matrix-matrix multiplication, matrix-vector multiplication, matrix addition, real scalar-matrix multiplication, respectively.

Independence of complex random variables $\mathbf{v}_m \in \mathbb{C}^{d_m}$ ($m = 1, \dots, M$) is defined as the independence of variables $\varphi_v(\mathbf{v}_m)$, i.e.,

$$I(\varphi_v(\mathbf{v}_1), \dots, \varphi_v(\mathbf{v}_M)) = 0, \quad (3.20)$$

where I stands for the mutual information and $\varphi_v(\mathbf{v}_m) \in \mathbb{R}^{2d_m}$ ($\forall m$). The entropy of a complex independent variable $\mathbf{v} \in \mathbb{C}^d$ is defined as

$$H(\mathbf{v}) = H(\varphi_v(\mathbf{v})). \quad (3.21)$$

3.3.2 Complex Independent Subspace Analysis

By the definition of independence for complex random variables detailed above, the complex valued ISA task [162] can be defined alike to the real case [(3.8)] as

$$\mathbf{x} = \mathbf{A} \mathbf{e}, \quad (3.22)$$

where $\mathbf{A} \in \mathbb{C}^{D \times D}$ is an unknown invertible mixing matrix, the hidden source \mathbf{e} is i.i.d. in time t and the $\mathbf{e}^m \in \mathbb{C}^{d_m}$ components of $\mathbf{e} = [\mathbf{e}_1; \dots; \mathbf{e}_M] \in \mathbb{C}^D$ ($D = \sum_{m=1}^M d_m$) are independent, i.e., $I(\varphi_v(\mathbf{e}_1), \dots, \varphi_v(\mathbf{e}_M)) = 0$. The goal is to estimate the mixing matrix \mathbf{A} (or its inverse) and the hidden source \mathbf{e} by making

use of the observations \mathbf{x} only.

3.3.3 Identification Method for Complex ISA

Now, we show that one can reduce the solution of the complex ISA model to a real ISA problem in case of certain a ‘non-Gaussian’ assumption. Namely, let us in addition assume in the complex ISA model that at most one of the random variable $\varphi_v(\mathbf{e}^m) \in \mathbb{R}^{2d_m}$ is Gaussian. Now, applying transformation φ_v to the complex ISA equation (Eq. (3.22)) and making use of the operation preserving properties of transformations φ_v, φ_M [see (3.17)], one gets:

$$\varphi_v(\mathbf{x}) = \varphi_M(\mathbf{A})\varphi_v(\mathbf{e}). \quad (3.23)$$

Given that (i) the independence of $\mathbf{e}^m \in \mathbb{C}^{d_m}$ is equivalent to that of $\varphi_v(\mathbf{e}^m) \in \mathbb{R}^{2d}$, and (ii) the existence of the inverse of $\varphi_M(\mathbf{A})$ is inherited from \mathbf{A} [see (3.15)], we end up with a real valued ISA task with observation $\varphi_v(\mathbf{x})$ and M pieces of $2d_m$ -dimensional hidden components $\varphi_v(\mathbf{e}^m)$. The consideration can also be extended to the non-i.i.d. case, for further details, see [163].

3.4 Nonparametric Models

The general ISA problem of separating sources with nonparametric dynamics has been hardly touched in the literature yet [110, 136]. [136] focused on the separation of stationary and ergodic source components of known and equal dimensions in case of constrained mixing matrices. [110] was dealing with wide sense stationary sources that (i) are supposed to be block-decorrelated for all time-shifts and (ii) have equal and known dimensional source components. The goal of this section is to extend ISA to the case of (i) nonparametric, asymptotically stationary source dynamics and (ii) unknown source component dimensions. Particularly, (i) we address the problem of ISA with nonparametric, asymptotically stationary dynamics, (ii) beyond this extension we also treat the case of unknown and possibly different source component dimensions, (iii) we allow the temporal evolution of the sources to be coupled; it is sufficient that their driving noises are independent and (iv) we

propose a simple estimation scheme by reducing the solution of the problem to kernel regression and ISA.

The structure of this section is as follows: Section 3.4.1 formulates the problem set-up. In Section 3.4.2 we describe our identification method.

3.4.1 Functional Autoregressive Independent Process Analysis

In this section we formally define the problem set-up [164]. In our framework we use functional autoregressive (fAR) processes to model nonparametric stochastic time series. Our goal is to develop dual estimation methods, i.e., to estimate both the system parameters and the hidden states for the functional autoregressive independent process analysis (fAR-IPA) model, which is defined as follows. Assume that the observation (\mathbf{x}) is a linear mixture (\mathbf{A}) of the hidden source (\mathbf{s}), which evolves according to an unknown fAR dynamics (\mathbf{f}) with independent driving noises (\mathbf{e}). Formally,

$$\mathbf{s}_t = \mathbf{f}(\mathbf{s}_{t-1}, \dots, \mathbf{s}_{t-L_s}) + \mathbf{e}_t, \quad (3.24)$$

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \quad (3.25)$$

where the unknown mixing matrix $\mathbf{A} \in \mathbb{R}^{D \times D}$ is invertible, L_s is the order of the process and the $\mathbf{e}^m \in \mathbb{R}^{d_m}$ components of $\mathbf{e} = [\mathbf{e}^1; \dots; \mathbf{e}^M] \in \mathbb{R}^D$ ($D = \sum_{m=1}^M d_m$) satisfy the ISA assumptions. The goal of the fAR-IPA problem is to estimate (i) the mixing matrix \mathbf{A} (or its inverse $\mathbf{W} = \mathbf{A}^{-1}$) and (ii) the original source \mathbf{s}_t by using observations \mathbf{x}_t only.

We list a few interesting special cases:

- If we knew the parametric form of \mathbf{f} , and if it were linear, then the problem would be the AR-IPA task [159].
- If we assume that the dynamics of the hidden layer is zero-order AR ($L_s = 0$), then the problem reduces to the original ISA problem [92].

3.4.2 Identification Method for fAR-IPA

We consider the dual estimation of the system described in (3.24)–(3.25). In what follows, we will propose a separation technique with which we can reduce the fAR-IPA estimation problem ((3.24)–(3.25)) to a functional AR process identification and an ISA problem. To obtain strongly consistent fAR estimation, the Nadaraya-Watson kernel regression technique is invoked.

More formally, the estimation of the fAR-IPA problem (3.24)–(3.25) can be accomplished as follows. The observation process \mathbf{x} is invertible linear transformation of the hidden fAR source process \mathbf{s}_t and thus it is also fAR process with innovation $\mathbf{A}\mathbf{e}_t$

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t = \mathbf{A}\mathbf{f}(\mathbf{s}_{t-1}, \dots, \mathbf{s}_{t-L_s}) + \mathbf{A}\mathbf{e}_t \quad (3.26)$$

$$= \mathbf{A}\mathbf{f}(\mathbf{A}^{-1}\mathbf{x}_{t-1}, \dots, \mathbf{A}^{-1}\mathbf{x}_{t-L_s}) + \mathbf{A}\mathbf{e}_t = \mathbf{g}(\mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-L_s}) + \mathbf{n}_t, \quad (3.27)$$

where function

$$\mathbf{g}(\mathbf{u}_1, \dots, \mathbf{u}_{L_s}) = \mathbf{A}\mathbf{f}(\mathbf{A}^{-1}\mathbf{u}_1, \dots, \mathbf{A}^{-1}\mathbf{u}_{L_s}) \quad (3.28)$$

describes the temporal evolution of \mathbf{x}_t , and

$$\mathbf{n}_t = \mathbf{A}\mathbf{e}_t \quad (3.29)$$

stands for the driving noise of the observation. Making use of this form, the fAR-IPA estimation can be carried out by fAR fit to observation \mathbf{x}_t followed by ISA on $\hat{\mathbf{n}}_t$, the estimated innovation of \mathbf{x}_t .

Note that Eq. (3.27) can be considered as a nonparametric regression problem; we have

$$\mathbf{u}_t = [\mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-L_s}], \quad \mathbf{v}_t = \mathbf{x}_t \quad (t = 1, \dots, T) \quad (3.30)$$

samples from the unknown relation

$$\mathbf{v}_t = \mathbf{g}(\mathbf{u}_t) + \mathbf{n}_t, \quad (3.31)$$

where \mathbf{u} , \mathbf{v} , and \mathbf{n} are the explanatory-, response variables and noise, respectively,

and \mathbf{g} is the unknown conditional mean or regression function. Nonparametric techniques can be applied to estimate the unknown mean function

$$\mathbf{g}(\mathbf{U}) = \mathbb{E}(\mathbf{V}|\mathbf{U}), \quad (3.32)$$

e.g., by carrying out kernel density estimation for random variables (\mathbf{u}, \mathbf{v}) and \mathbf{u} , where \mathbb{E} stands for expectation. The resulting Nadaraya-Watson estimator (i) takes the simple form

$$\hat{\mathbf{g}}_0(\mathbf{u}) = \frac{\sum_{t=1}^T \mathbf{v}_t K\left(\frac{\mathbf{u}-\mathbf{u}_t}{h}\right)}{\sum_{t=1}^T K\left(\frac{\mathbf{u}-\mathbf{u}_t}{h}\right)}, \quad (3.33)$$

where K and $h > 0$ denotes the applied kernel (a non-negative real-valued function that integrates to one) and bandwidth, respectively. It can be used to provide a strongly consistent estimation of the regression function \mathbf{g} for stationary \mathbf{x}_t processes [165]. It has been shown recently [166] that for first order and only *asymptotically stationary* fAR processes, under mild regularity conditions, one can get strongly consistent estimation for the innovation \mathbf{n}_t by applying the recursive version of the Nadaraya-Watson estimator

$$\hat{\mathbf{g}}(\mathbf{u}) = \frac{\sum_{t=1}^T t^{\beta D} \mathbf{v}_t K(t^{\beta}(\mathbf{u} - \mathbf{u}_t))}{\sum_{t=1}^T t^{\beta D} K(t^{\beta}(\mathbf{u} - \mathbf{u}_t))}, \quad (3.34)$$

where the bandwidth is parameterized by $\beta \in (0, 1/D)$.

3.5 Convolutional Models

In this section we address the blind subspace deconvolution (BSSD) problem; an the extension of both the blind source deconvolution and the independent subspace analysis tasks. One can think of the BSSD problem as a cocktail party with groups, held in an echoic room. For the undercomplete case, where we have ‘more microphones than sources’, it has been shown recently that the problem can be reduced to ISA by means of temporal concatenation [123]. However, the associated ISA problem can easily become ‘high dimensional’. The dimensionality problem can be circumvented by applying a linear predictive approximation (LPA) based reduction [167]. Here, we show that it is possible to extend the LPA idea to the *complete*

BSSD task, where the number of ‘microphones’ equals to the number of ‘sources’.³ In the undercomplete case, the LPA based solution was based on the observation that the polynomial matrix describing the temporal convolution had, under rather general conditions⁴, a polynomial matrix left inverse. In the complete case such an inverse doesn’t exist in general. However, provided that the convolution can be represented by an infinite order autoregressive process, one can construct an efficient estimation method for the hidden components via an asymptotically consistent LPA procedure. This thought is used here to extend the technique of [167] to the complete case.

The section is structured as follows: in Section 3.5.1 we define the complete blind subspace deconvolution problem, we detail our solution technique in Section 3.5.2.

3.5.1 Complete Blind Subspace Deconvolution

Here, we define the BSSD task [123]. Assume that we have M hidden, independent, multidimensional *components* (random variables). Suppose also that only their

$$\mathbf{x}_t = \sum_{l=0}^{L_e} \mathbf{H}_l \mathbf{e}_{t-l} \quad (3.35)$$

convolutive mixture is available for observation, where $\mathbf{x}_t \in \mathbb{R}^{D_x}$ and \mathbf{e}_t is the concatenation of the components $\mathbf{e}_t^m \in \mathbb{R}^{d_m}$, that is $\mathbf{e}_t = [\mathbf{e}_t^1; \dots; \mathbf{e}_t^M] \in \mathbb{R}^{D_e}$ ($D_e = \sum_{m=1}^M d_m$). By describing the convolution using the polynomial matrix $\mathbf{H}[z] = \sum_{l=0}^{L_e} \mathbf{H}_l z^l \in \mathbb{R}[z]^{D_x \times D_e}$, one may write Eq. (3.35) compactly as

$$\mathbf{x} = \mathbf{H}[z]\mathbf{e}. \quad (3.36)$$

We assume that the components \mathbf{e}^m fulfill the ISA assumptions. The goal of the BSSD problem is to estimate the original source \mathbf{e}_t by using observations \mathbf{x}_t only. While $D_x > D_e$ is the *undercomplete* case, $D_x = D_e$ is the *complete* one. The case

³The overcomplete BSSD task is challenging and as of yet no general solution is known.

⁴If the coefficients of the undercomplete polynomial matrix are drawn from a non-degenerate continuous distribution, such an inverse exists with probability one [168].

$L_e = 0$ corresponds to the ISA task, and if $d_m = 1$ ($\forall m$) also holds, then the ICA task is recovered. In the BSD task $d_m = 1$ ($\forall m$) and L_e is a non-negative integer.

Contrary to previous works [123, 167] focusing on the undercomplete BSSD problem, here [169] we address the complete task ($D = D_x = D_e$). In the complete BSSD problem we assume that the polynomial matrix $\mathbf{H}[z]$ is stable.

3.5.2 Identification Method for Complete BSSD

Below, we derive our separation principle based solution method for the complete BSSD problem.

The invertibility of $\mathbf{H}[z]$ implies that the observation process \mathbf{x} can be represented as an infinite order autoregressive (AR) process [170]:

$$\mathbf{x}_t = \sum_{j=1}^{\infty} \mathbf{F}_j \mathbf{x}_{t-j} + \mathbf{F}_0 \mathbf{e}_t. \quad (3.37)$$

By applying a finite order LPA approximation (fitting an AR process to \mathbf{x}), the innovation process $\mathbf{F}_0 \mathbf{e}_t$ can be estimated. The innovation can be seen as the observation of an ISA problem because components of \mathbf{e} are independent: ISA techniques can be used to identify components \mathbf{e}^m . Choosing the order of the fitted AR process to \mathbf{x} as $p = o(T^{\frac{1}{3}}) \xrightarrow{T \rightarrow \infty} \infty$, where T denotes the number of samples, guarantees that the AR approximation for the MA (moving average) model is asymptotically consistent [171].

3.6 Information Theoretical Estimations via Random Projections

The estimation of relevant information theoretical quantities, such as entropy, mutual information, and various divergences is computationally expensive in high dimensions. However, consistent estimation of these quantities is possible by nearest neighbor (NN) methods (see, e.g., [172]) that use the pairwise distances of sample points. Although search for nearest neighbors can also be expensive in high dimensions [173], low dimensional approximate isometric embedding of points of

high dimensional Euclidean space can be addressed by the Johnson-Lindenstrauss Lemma [174] and the related random projection (RP) methods [175, 176]. The RP approach proved to be successful, e.g., in classification, clustering, search for *approximate NN* (ANN), dimension estimation of manifolds, estimation of mixture of Gaussian models, compressions, data stream computation (see, e.g., [177]). We note that the RP approach is also related to compressed sensing [178].

In this section [179] we show a novel application of the RP technique: we estimate information theoretical quantities using the ANN-preserving properties of the RP technique. We present our RP based approach through the ISA problem. The ISA task can be viewed as the minimization of the mutual information between the estimated components, or equivalently as the minimization of the sum of Shannon’s multidimensional differential entropies of the estimated components on the orthogonal group [180]:

$$J(\mathbf{W}) = \sum_{m=1}^M H(\mathbf{y}^m) \rightarrow \min_{\mathbf{W} \in \mathcal{O}^D}, \quad (3.38)$$

where

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \quad \mathbf{y} = [\mathbf{y}^1; \dots; \mathbf{y}^M], \quad \mathbf{y}^m \in \mathbb{R}^{d_m} \quad (3.39)$$

and d_m s are given. Estimation of cost function J however involves multidimensional entropy estimation, which is computationally expensive in high dimensions, but can be executed by NN methods consistently [172]. It has been shown in [181] (in the field of image registration with high dimensional features) that the computational load can be decreased somewhat by

- dividing the samples into groups and then
- computing the averages of the group estimates.

We will combine this *parallelizable ensemble approach* with the ANN-preserving properties of RPs and get drastic savings. We suggest the following entropy estimation method⁵, for each estimated ISA component $\mathbf{v} := \hat{\mathbf{y}}_{\text{ISA}}^m$:

⁵The idea can be used for a number of information theoretical quantities, provided that they can be estimated by means of pairwise Euclidean distances of the samples.

- divide the T samples $\{\mathbf{v}(1), \dots, \mathbf{v}(T)\}$ into N groups indexed by sets I_1, \dots, I_N so that each group contains K samples,
- for all fixed groups take the random projection of \mathbf{v} as

$$\mathbf{v}_{n,\text{RP}}(t) := \mathbf{R}_n \mathbf{v}(t) \quad (t \in I_n; n = 1, \dots, N; \mathbf{R}_n \in \mathbb{R}^{d_m \times d_m}), \quad (3.40)$$

- average the estimated entropies of the RP-ed groups to get the estimation

$$\hat{H}(\mathbf{v}) = \frac{1}{N} \sum_{n=1}^N \hat{H}(\mathbf{v}_{n,\text{RP}}). \quad (3.41)$$

Our particular choice for \mathbf{R}_n can be found in Section 5.3.6.

Chapter 4

Numerical Experiments – Group-Structured Dictionary Learning

In this chapter we demonstrate the efficiency of structured codes. For illustration purposes we chose the online group-structured dictionary learning approach. The efficiency of the method is presented in 3 different applications: inpainting of natural images (Section 4.1), structured non-negative matrix factorization of faces (Section 4.2) and collaborative filtering (Section 4.3).

4.1 Inpainting of Natural Images

We studied the following issues on natural images:

1. Is structured dictionary \mathbf{D} beneficial for inpainting of patches of natural images, and how does it compare to the dictionary of classical sparse representation? During learning of \mathbf{D} , training samples \mathbf{x}_i were fully observed (i.e., $\Delta_i = \mathbf{I}$).
2. In this inpainting problem of image patches, we also studied the case when the training samples \mathbf{x}_i were partially observed ($\Delta_i \neq \mathbf{I}$).



Figure 4.1: Illustration of the used natural image dataset. (a): 12 images of similar kind were used to select patches for the training \mathbf{X}_{tr} , validation \mathbf{X}_{val} , and test \mathbf{X}_{test} sets. (b): test image used for the illustration of full image inpainting.

3. We also show results for inpainting of *full images* using a dictionary learned from partially observed ($\Delta_i \neq \mathbf{I}$) patches.

In our numerical experiments we used $\mathcal{D}_i = S_2^{d_x}(\forall i)$, $\mathcal{A} = \mathbb{R}^{d_\alpha}$ without additional weighing ($\mathbf{A}^G = \mathbf{I}$, $\forall G \in \mathcal{G}$). Group structure \mathcal{G} of vector α was realized on a 16×16 torus ($d_\alpha = 256$) with $|\mathcal{G}| = d_\alpha$ applying $r = 0, 1, 2$, or 3 neighbors to define \mathcal{G} . For $r = 0$ ($\mathcal{G} = \{\{1\}, \dots, \{d_\alpha\}\}$) the classical sparse representation is recovered. Our test database was the ICA natural image database.¹ We chose 12 of the 13 images of the dataset to study the first two questions above (see Fig. 4.1(a)), and used the 13th picture for studying the third question (Fig. 4.1(b)). For each of the 12 images, we sampled $131,072 = 2^{17}$ pieces of 8×8 disjunct image patches randomly (without replacement). This patch set was divided to a training set \mathbf{X}_{tr} made of 65,536 pieces, and to a validation (\mathbf{X}_{val}) and test (\mathbf{X}_{test}) set with set sizes 32,768. Each patch was normalized to zero average and unit ℓ_2 -norm.

In the **first experiment** \mathbf{x}_i s were fully observed ($\Delta_i = \mathbf{I}$) and thus the update of their statistics was precise. This is called the BCD case in the figures. Matrix \mathbf{D} was learned on the set \mathbf{X}_{tr} , columns \mathbf{d}_j were initialized by using a uniform distribution on the surface of the ℓ_2 -sphere. Pixels of the \mathbf{x} patches in the validation and test sets were removed with probability p_{test}^{val} . For a given noise-free image patch \mathbf{x} , let \mathbf{x}_O denote its observed version, where O stands for the indices of the available coordinates. The task was the inpainting of the missing pixels of \mathbf{x} by means of the pixels present (\mathbf{x}_O) and by the learned matrix \mathbf{D} . After removing the rows of \mathbf{D} corresponding to missing pixels of \mathbf{x} , the resulting \mathbf{D}_O and \mathbf{x}_O were used to estimate α . The final estimation of \mathbf{x} was $\hat{\mathbf{x}} = \mathbf{D}_O \alpha$. According to preliminary

¹ See <http://www.cis.hut.fi/projects/ica/data/images/>.

experiments, learning rate ρ and mini-batch size R were set to 32 and 64 respectively (the estimation was robust as a function of ρ and R). In the updates of \mathbf{z} and $\boldsymbol{\alpha}$ (2.24) only minor changes were experienced after 2-3 iterations, thus the number of iterations T_α was set to 5. Concerning the other parameters, we used $\eta = 0.5$, and $\kappa \in \{2^{-19}, 2^{-18}, \dots, 2^{-10}\}$. The ϵ smoothing parameter was 10^{-5} , and the iteration number for the update of \mathbf{D} was $T_D = 5$. Values of p_{test}^{val} were chosen from set $\{0.3, 0.5, 0.7, 0.9\}$, so for the case of $p_{test}^{val} = 0.9$, only 10% of the pixels of \mathbf{x} were observed. For each fixed neighborhood size r and parameter p_{test}^{val} , κ was chosen as the minimum of mean squared error (MSE) using \mathbf{D} trained on patch set \mathbf{X}_{tr} and evaluated on \mathbf{X}_{val} . Having found this optimal κ on the validation set, we used its value to compute the MSE on \mathbf{X}_{test} . Then we changed the roles of \mathbf{X}_{val} and \mathbf{X}_{test} , that is, validated on \mathbf{X}_{test} , and tested on \mathbf{X}_{val} . This procedure was repeated for four random initializations (\mathbf{D}_0) and different corruptions (\mathbf{X}_{val} , \mathbf{X}_{test}). The average MSE values (multiplied by 100) and their standard deviations for different neighbor sizes r and corruption rates p_{test}^{val} are summarized in Table 4.1. This table shows that (i) the inpainting error grows with the corruption rate p_{test}^{val} , (ii) compared to sparse representation ($r = 0$) small neighborhood size $r = 1$ gives rise to similar results, $r = 2$ is better and $r = 3$ seems to be the best for all cases with 13 – 19% improvement in precision for MSE. Learned and average quality dictionaries \mathbf{D} can be seen in Fig. 4.2 ($r = 0$ no structure, $r = 2, 3$ with torus structure). Based on this experiment we can conclude that the structured algorithm gives rise to better results than ordinary sparse representations.

In the **second experiment**, the size of the neighborhood was fixed, set to $r = 3$. We learned dictionary \mathbf{D} on *partially observed* patches ($\Delta_i \neq \mathbf{I}$). The probability p_{tr} of missing any pixel from the observations in the training set assumed values from the set $\{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$. In this case, we updated \mathbf{e} using the approximation Eq. (2.32), hence we call this method approximate-BCD (or BCDA, for short). The other experimental details were identical to the previous case (i.e., when $\Delta_i = \mathbf{I}$). Results and statistics for MSE are provided for a smaller (0.3) and for a larger (0.7) value of p_{test}^{val} in Table 4.2 for different probability values p_{tr} . We found that increasing p_{tr} up to $p_{tr} = 0.7$ MSE values grow slowly. Note that we kept the number of samples \mathbf{x}_i at 65,536 identical to the previous case ($\Delta_i = \mathbf{I}$), and thus by increasing p_{tr} the effective number of observations/coordinates decreases.

Table 4.1: BCD: $100 \times$ the MSE average (\pm std) as a function of neighbors ($r = 0$: sparse representation, no structure) for different p_{test}^{val} corruption rates.

	$p_{test}^{val} = 0.3$	$p_{test}^{val} = 0.5$
$r = 0$	0.65 (± 0.002)	0.83 (± 0.003)
$r = 1$	0.60 (± 0.005 ; +6.78%)	0.85 (± 0.017 ; -2.25%)
$r = 2$	0.59 (± 0.005 ; +10.39%)	0.81 (± 0.008 ; +2.67%)
$r = 3$	0.56 (± 0.002 ; + 16.38%)	0.71 (± 0.002 ; + 16.01%)
	$p_{test}^{val} = 0.7$	$p_{test}^{val} = 0.9$
$r = 0$	1.10 (± 0.002)	1.49 (± 0.006)
$r = 1$	1.10 (± 0.029 ; +0.27%)	1.45 (± 0.004 ; +2.96%)
$r = 2$	1.12 (± 0.029 ; -1.09%)	1.46 (± 0.029 ; +2.51%)
$r = 3$	0.93 (± 0.001 ; + 18.93%)	1.31 (± 0.002 ; + 13.87%)

Table 4.2: BCDA ($r = 3$): $100 \times$ the MSE average (\pm std) for different for different p_{test}^{val} and p_{tr} corruption rates.

	$p_{tr} = 0$	$p_{tr} = 0.1$	$p_{tr} = 0.3$
$p_{test}^{val} = 0.3$	0.55 (± 0.003)	0.56 (± 0.001)	0.57 (± 0.003)
$p_{test}^{val} = 0.7$	0.91 (± 0.002)	0.91 (± 0.002)	0.91 (± 0.002)
	$p_{tr} = 0.5$	$p_{tr} = 0.7$	$p_{tr} = 0.9$
$p_{test}^{val} = 0.3$	0.59 (± 0.001)	0.61 (± 0.002)	0.71 (± 0.007)
$p_{test}^{val} = 0.7$	0.92 (± 0.003)	0.93 (± 0.002)	0.96 (± 0.003)

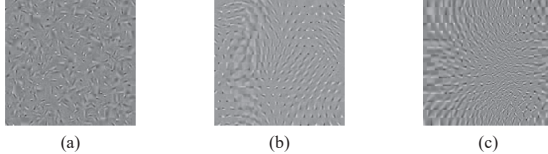


Figure 4.2: Illustration of the online learned group-structured \mathbf{D} dictionaries with the BCD technique and MSE closest to the average (see Table 4.1) and $p_{test}^{val} = 0.7$. (a): $r = 0$, (b): $r = 2$, (c): $r = 3$.

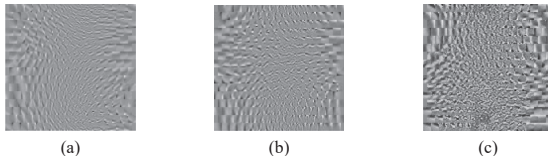


Figure 4.3: Illustration of the online learned group-structured \mathbf{D} dictionaries for the BCDA technique with MSE closest to the average (see Table 4.2) and $p_{test}^{val} = 0.7$. (a): $p_{tr} = 0$, (b): $p_{tr} = 0.1$, (c): $p_{tr} = 0.5$.

Learned average quality dictionaries \mathbf{D} are shown in Fig. 4.3 for $p_{test}^{val} = 0.7$. Note that the MSE values are still relatively small for missing pixel probability $p_{tr} = 0.9$ ($100 \times$ MSE maximum is about 0.96), thus our proposed method is still efficient in this case. Reconstruction with value 0.92 ($100 \times$ MSE) is shown in Fig. 4.4.

In our **third illustration** we show full image inpainting using dictionary \mathbf{D} learned with $p_{tr} = 0.5$ and using the 13th image (\mathbf{X}) shown in Fig. 4.1(b). We executed inpainting consecutively on all 8×8 patches of image \mathbf{X} and for each pixel of image \mathbf{X} , we averaged all estimations $\hat{\mathbf{x}}_i$ from all 8×8 patches that contained the pixel. Results are shown in Fig. 4.4 for $p_{test}^{val} = 0.3$ and 0.7 values. We also provide the PSNR (peak signal-to-noise ratio) values of our estimations. This measure for vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ (i.e., for vectors formed from the pixels of the image) is defined as

$$PSNR(\mathbf{u}, \mathbf{v}) = 10 \log_{10} \left[\frac{(\max(\max_i |u_i|, \max_j |v_j|))^2}{\frac{1}{d} \|\mathbf{u} - \mathbf{v}\|_2^2} \right], \quad (4.1)$$

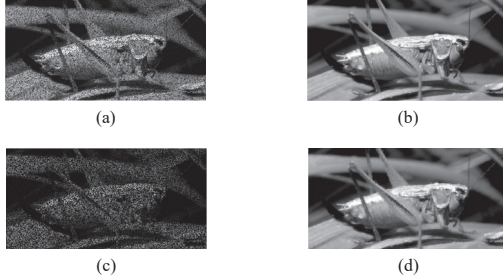


Figure 4.4: Inpainting illustration using the online learned group-structured D dictionaries for the BCDA technique with MSE closest to the average (see Table 4.2) and $p_{tr} = 0.5$. (a): measured, (b): estimated, PSNR = 36 dB. (a)-(b): $p_{test}^{val} = 0.3$. (c)-(d): the same as (a)-(b), but with $p_{test}^{val} = 0.7$, in (d) PSNR = 29 dB.

where the higher value is the better. Acceptable values in wireless transmission (lossy image and video compression) are around 20 – 25 dB (30 dB). By means of D and for missing probability $p_{test}^{val} = 0.3$ we achieved 36 dB PSNR, whereas for missing probability $p_{test}^{val} = 0.7$ we still have 29 dB PSNR, underlining the efficiency of our method.

4.2 Online Structured Non-negative Matrix Factorization on Faces

It has been shown on the CBCL database that dictionary vectors (\mathbf{d}_i) of the offline NMF method can be interpreted as face components [94]. However, to the best of our knowledge, there is no existing NMF algorithm as of yet, which could handle general \mathcal{G} group structures in an online fashion. Our OSDL method is able to do that, can also cope with only partially observed inputs, and can be extended with non-convex sparsity-inducing norms. We illustrate our approach on the color FERET² dataset: we set $\mathcal{D}_i = \mathcal{S}_2^{d_x} \cap \mathbb{R}_+^{d_x}$ ($\forall i$), $\mathcal{A} = \mathbb{R}_+^{d_x}$, $\Delta_i = \mathbf{I}$ and $\eta = 0.5$. We selected 1,736 facial pictures from the dataset. Using affine transformations we positioned

²See <http://face.nist.gov/colorferet/>.

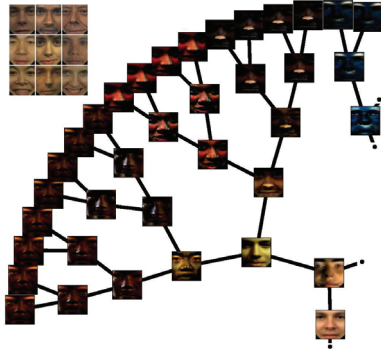


Figure 4.5: Illustration of the online learned structured NMF dictionary. Upper left corner: training samples.

the noses and eyes to the same pixel coordinates, reduced the image sizes to 140×120 , and set their l_2 norms to be one. These images were the observations for our ODSL method ($\mathbf{x}_i, d_x = 49, 140 = 140 \times 120 \times 3$ minus some masking). The group structure \mathcal{G} was chosen to be hierarchical; we applied a full, 8-level binary tree. Each node with its corresponding descendants formed the sets of $G \in \mathcal{G}$ ($d_\alpha = 255$). According to our experiments, the learned dictionary \mathbf{D} was influenced mostly by the constant κ , and similarly to Section 4.1, it proved to be quite insensitive to the value of the learning factor ρ , and to the size of the mini-batches (R). Fig. 4.5 shows a few elements from the online estimated structured NMF dictionary (using $\kappa = 2^{-10.5}$, $\rho = 32$, $R = 8$, $\mathbf{A}^G = \mathbf{I} \ (\forall G \in \mathcal{G})$, $T_\alpha = 5$, $T_D = 5$ and $\varepsilon = 10^{-5}$). We can observe that the proposed algorithm is able to naturally develop and hierarchically organize the elements of the dictionary: towards the leaves the learned filters reveal more and more details. We can also notice that the colors are separated as well. This example demonstrates that our method can be used for large problems where the dimension of the observations is about 50,000.

4.3 Collaborative Filtering

The proliferation of online services and the thriving electronic commerce overwhelms us with alternatives in our daily lives. To handle this information overload and to help users in efficient decision making, recommender systems (RS) have been designed. The goal of RSs is to recommend personalized items for online users when they need to choose among several items. Typical problems include recommendations for which movie to watch, which jokes/books/news to read, which hotel to stay at, or which songs to listen to.

One of the most popular approaches in the field of recommender systems is *collaborative filtering* (CF). The underlying idea of CF is very simple: Users generally express their tastes in an explicit way by rating the items. CF tries to estimate the users' preferences based on the ratings they have already made on items and based on the ratings of other, similar users. For a recent review on recommender systems and collaborative filtering, see e.g., [80].

Novel advances on CF show that *dictionary learning* based approaches can be efficient for making predictions about users' preferences [182]. The dictionary learning based approach assumes that (i) there is a latent, unstructured feature space (hidden representation) behind the users' ratings, and (ii) a rating of an item is equal to the product of the item and the user's feature. To increase the generalization capability, usually ℓ_2 regularization is introduced both for the dictionary and for the users' representation.

Here, we extend the application domain of structured dictionary learning in the direction of collaborative filtering. With respect to CF, further constraints appear for structured dictionary learning since (i) online learning is desired and (ii) missing information is typical. There are good reasons for them: novel items/users may appear and user preferences may change over time. Adaptation to users also motivate online methods. Furthermore, users can evaluate only a small portion of the available items, which leads to incomplete observations, missing rating values.

To do so, we formulate the CF task as an OSDL optimization problem in Section 4.3.1. According to the CF literature, oftentimes neighbor-based corrections improve the precision of the estimation. We also use this technique (Section 4.3.2) to improve the OSDL estimations. Numerical results are presented in Section 4.3.3.

4.3.1 Collaborative Filtering as Structured Dictionary Learning

Below, the CF task is transformed into an OSDL problem. Consider the t^{th} user's known ratings as OSDL observations \mathbf{x}_{O_t} . Let the optimized group-structured dictionary on these observations be \mathbf{D} . Now, assume that we have a test user and his/her ratings, i.e., $\mathbf{x}_O \in \mathbb{R}^{|O|}$. The task is to estimate $\mathbf{x}_{\{1, \dots, d_x\} \setminus O}$, that is, the missing coordinates of \mathbf{x} (the missing ratings of the user) that can be accomplished as follows:

1. Remove the rows of the non-observed $\{1, \dots, d_x\} \setminus O$ coordinates from \mathbf{D} . The obtained $|O| \times d_\alpha$ sized matrix \mathbf{D}_O and \mathbf{x}_O can be used to estimate α by solving the structured sparse coding problem (2.2).
2. Using the estimated representation α , estimate \mathbf{x} as

$$\hat{\mathbf{x}} = \mathbf{D}\alpha. \quad (4.2)$$

4.3.2 Neighbor Based Correction

According to the CF literature, neighbor based correction schemes may further improve the precision of the estimations [80]. This neighbor correction approach

- relies on the assumption that similar items (e.g., jokes/movies) are rated similarly and
- it can be adapted to OSDL-based CF estimation in a natural fashion.

Here, we detail the idea. Let us assume that the similarities $s_{ij} \in \mathbb{R}$ ($i, j \in \{1, \dots, d_x\}$) between individual items are given. We shall provide similarity forms in Section 4.3.3. Let $\mathbf{d}_k \alpha_t \in \mathbb{R}$ be the OSDL estimation for the rating of the k^{th} non-observed item of the t^{th} user ($k \notin O_t$), where $\mathbf{d}_k \in \mathbb{R}^{1 \times d_\alpha}$ is the k^{th} row of matrix $\mathbf{D} \in \mathbb{R}^{d_x \times d_\alpha}$, and $\alpha_t \in \mathbb{R}^{d_\alpha}$ is computed according to Section 4.3.1.

Let the prediction error on the observable item neighbors (j) of the k^{th} item of the t^{th} user ($j \in O_t \setminus \{k\}$) be $\mathbf{d}_j \alpha_t - x_{jt} \in \mathbb{R}$. These prediction errors can be used for the correction of the OSDL estimation ($\mathbf{d}_k \alpha_t$) by taking into account the s_{ij}

similarities:

$$\hat{x}_{kt} = \mathbf{d}_k \boldsymbol{\alpha}_t + \gamma_1 \left[\frac{\sum_{j \in O_t \setminus \{k\}} s_{kj} (\mathbf{d}_j \boldsymbol{\alpha}_t - x_{jt})}{\sum_{j \in O_t \setminus \{k\}} s_{kj}} \right], \text{ or} \quad (4.3)$$

$$\hat{x}_{kt} = \gamma_0 (\mathbf{d}_k \boldsymbol{\alpha}_t) + \gamma_1 \left[\frac{\sum_{j \in O_t \setminus \{k\}} s_{kj} (\mathbf{d}_j \boldsymbol{\alpha}_t - x_{jt})}{\sum_{j \in O_t \setminus \{k\}} s_{kj}} \right], \quad (4.4)$$

where $k \notin O_t$. Here, (4.3) is analogous to the form of [182], (4.4) is a simple modification: it modulates the first term with a separate γ_0 weight.

4.3.3 Numerical Results

This section is structured as follows: We have chosen the Jester dataset for the illustration of the OSDL based CF approach. It is a standard benchmark for CF. This is what we introduce first. Then we present our preferred item similarities. The performance measure used to evaluate the CF based estimation follows. The final part of this section is about our numerical experiences.

The Jester Dataset

The dataset [183] contains 4,136,360 ratings from 73,421 users to 100 jokes on a continuous $[-10, 10]$ range. The worst and best possible gradings are -10 and $+10$, respectively. A fixed 10 element subset of the jokes is called gauge set and it was evaluated by all users. Two third of the users have rated at least 36 jokes, and the remaining ones have rated between 15 and 35 jokes. The average number of user ratings per joke is 46.

Item Similarities

In the neighbor correction step (4.3) or (4.4) we need the s_{ij} values representing the similarities of the i^{th} and j^{th} items. We define this value as the similarity of the i^{th}

and j^{th} rows (\mathbf{d}_i and \mathbf{d}_j) of the optimized OSDL dictionary \mathbf{D} [182]:

$$S_1: s_{ij} = s_{ij}(\mathbf{d}_i, \mathbf{d}_j) = \left(\frac{\max(0, \langle \mathbf{d}_i, \mathbf{d}_j \rangle)}{\|\mathbf{d}_i\|_2 \|\mathbf{d}_j\|_2} \right)^\beta, \text{ or} \quad (4.5)$$

$$S_2: s_{ij} = s_{ij}(\mathbf{d}_i, \mathbf{d}_j) = \left(\frac{\|\mathbf{d}_i - \mathbf{d}_j\|_2^2}{\|\mathbf{d}_i\|_2 \|\mathbf{d}_j\|_2} \right)^{-\beta}, \quad (4.6)$$

where $\beta > 0$ is the parameter of the similarity measure. Quantities s_{ij} are non-negative; if the value of s_{ij} is close to zero (large) then the i^{th} and j^{th} items are very different (very similar).

Performance Measure

In our numerical experiments we used the RMSE (root mean square error) measure for the evaluation of the quality of the estimation, since RMSE is one of most popular measures in the CF literature. The RMSE measure is the average squared difference of the true and the estimated rating values:

$$RMSE = \sqrt{\frac{1}{|\mathcal{S}|} \sum_{(i,t) \in \mathcal{S}} (x_{it} - \hat{x}_{it})^2}, \quad (4.7)$$

where \mathcal{S} denotes either the validation or the test set.

Evaluation

Here we illustrate the efficiency of the OSDL-based CF estimation on the Jester database using the RMSE performance measure. To the best of our knowledge, the top results on this database are RMSE = 4.1123 [184] and RMSE = 4.1229 [182]. Both works are from the same authors. The method in the first paper is called item neighbor and it makes use of only neighbor information. In [182], the authors used a bridge regression based unstructured dictionary learning model—with a neighbor correction scheme—, they optimized the dictionary by gradient descent and set d_α to 100. These are our performance baselines.

To study the capability of the OSDL approach in CF, we focused on the following issues:

- Is structured dictionary \mathbf{D} beneficial for prediction purposes, and how does it compare to the dictionary of classical (unstructured) sparse dictionary?
- How does the OSDL parameters and the similarity/neighbor correction applied affect the efficiency of the prediction?
- How do different group structures \mathcal{G} fit to the CF task?

In our numerical studies we chose the Euclidean unit sphere for $\mathcal{D}_i = S_2^{d_x} (\forall i)$, and $\mathcal{A} = \mathbb{R}^{d_\alpha}$, and no additional weighting was applied ($\mathbf{d}^G = \chi_G, \forall G \in \mathcal{G}$, where χ is the indicator function). We set η of the group-structured regularizer Ω to 0.5. Group structure \mathcal{G} of vector α was realized on

- a $d \times d$ toroid ($d_\alpha = d^2$) with $|\mathcal{G}| = d_\alpha$ applying $r \geq 0$ neighbors to define \mathcal{G} . For $r = 0$ ($\mathcal{G} = \{\{1\}, \dots, \{d_\alpha\}\}$) the classical sparse representation based dictionary is recovered.
- a hierarchy with a complete binary tree structure. In this case:
 - $|\mathcal{G}| = d_\alpha$, and group G of α_i contains the i^{th} node and its descendants on the tree, and
 - the size of the tree is determined by the number of levels l . The dimension of the hidden representation is then $d_\alpha = 2^l - 1$.

The size R of mini-batches was set either to 8, or to 16 and the forgetting rate ρ was chosen from set $\{0, \frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\}$. The κ weight of structure inducing regularizer Ω was chosen from the set $\{\frac{1}{2^{-1}}, \frac{1}{2^0}, \frac{1}{2^1}, \frac{1}{2^2}, \frac{1}{2^4}, \frac{1}{2^6}, \dots, \frac{1}{2^{14}}\}$. We studied similarities S_1, S_2 [see (4.5)-(4.6)] with both neighbor correction schemes [(4.3)-(4.4)]. In what follows, corrections based on (4.3) and (4.4) will be called S_1^0, S_2^0 and S_1, S_2 , respectively. Similarity parameter β was chosen from the set $\{0.2, 1, 1.8, 2.6, \dots, 14.6\}$. In the BCD step of the optimization of \mathbf{D} , $T_\alpha = 5$ iterations were applied. In the α optimization step, we used $T_D = 5$ iterations, whereas smoothing parameter ϵ was 10^{-5} .

We used a 90%–10% random split for the observable ratings in our experiments, similarly to [182]:

- training set (90%) was further divided into 2 parts:

- we chose the 80% observation set $\{O_t\}$ randomly, and optimized \mathbf{D} according to the corresponding \mathbf{x}_{O_t} observations,
- we used the remaining 10% for validation, that is for choosing the optimal OSDL parameters (r or l , κ , ρ), BCD optimization parameter (R), neighbor correction (S_1 , S_2 , S_1^0 , S_2^0), similarity parameter (β), and correction weights (γ_t s in (4.3) or (4.4)).
- we used the remaining 10% of the data for testing.

The optimal parameters were estimated on the validation set, and then used on the test set. The resulting RMSE score was the performance of the estimation.

Toroid Group Structure. In this section we provide results using toroid group structure. We set $d = 10$. The size of the toroid was 10×10 , and thus the dimension of the representation was $d_\alpha = 100$.

In the **first experiment** we study how the size of neighborhood (r) affects the results. This parameter corresponds to the ‘smoothness’ imposed on the group structure: when $r = 0$, then there is no relation between the \mathbf{d}_j columns in \mathbf{D} (no structure). As we increase r , the \mathbf{d}_j feature vectors will be more and more aligned in a smooth way. To this end, we set the neighborhood size to $r = 0$ (no structure), and then increased it to 1, 2, 3, 4, and 5. For each (κ, ρ, β) , we calculated the RMSE of our estimation, and then for each fixed (κ, ρ) pair, we minimized these RMSE values in β . The resulting validation and test surfaces are shown in Fig. 4.6. For the best (κ, ρ) pair, we also present the RMSE values as a function of β (Fig. 4.7). In this illustration we used S_1^0 neighbor correction and $R = 8$ mini-batch size. We note that we got similar results using $R = 16$ too. Our results can be summarized as follows.

- For a fixed neighborhood parameter r , we have that:
 - The validation and test surfaces are very similar (see Fig. 4.6(e)-(f)). It implies that the validation surfaces are good indicators for the test errors. For the best r , κ and ρ parameters, we can observe that the validation and test curves (as functions of β) are very similar. This is demonstrated in Fig. 4.7, where we used $r = 4$ neighborhood size and S_1^0 neighbor

correction. We can also notice that (i) both curves have only one local minimum, and (ii) these minimum points are close to each other.

- The quality of the estimation depends mostly on the κ regularization parameter. As we increase r , the best κ value is decreasing.
- The estimation is robust to the different choices of forgetting factors (see Fig. 4.6(a)-(e)). In other words, this parameter ρ can help in fine-tuning the results.
- Structured dictionaries ($r > 0$) are advantageous over those methods that do not impose structure on the dictionary elements ($r = 0$). For S_1^0 and S_2^0 neighbor corrections, we summarize the RMSE results in Table 4.3. Based on this table we can conclude that in the studied parameter domain
 - the estimation is robust to the selection of the mini-batch size (R). We got the best results using $R = 8$. Similarly to the role of parameter ρ , adjusting R can be used for fine-tuning.
 - the S_1^0 neighbor correction lead to the smallest RMSE value.
 - When we increase r up to $r = 4$, the results improve. However, for $r = 5$, the RMSE values do not improve anymore; they are about the same that we have using $r = 4$.
 - The smallest RMSE we could achieve was 4.0774, and the best known result so far was RMSE = 4.1123 [184]. This proves the efficiency of our OSDL based collaborative filtering algorithm.
 - We note that our RMSE result seems to be significantly better than the that of the competitors: we repeated this experiment 5 more times with different randomly selected training, test, and validation sets, and our RMSE results have never been worse than 4.08.

In the **second experiment** we studied how the different neighbor corrections (S_1 , S_2 , S_1^0 , S_2^0) affect the performance of the proposed algorithm. To this end, we set the neighborhood parameter to $r = 4$ because it proved to be optimal in the previous experiment. Our results are summarized in Table 4.4. From these results we can observe that

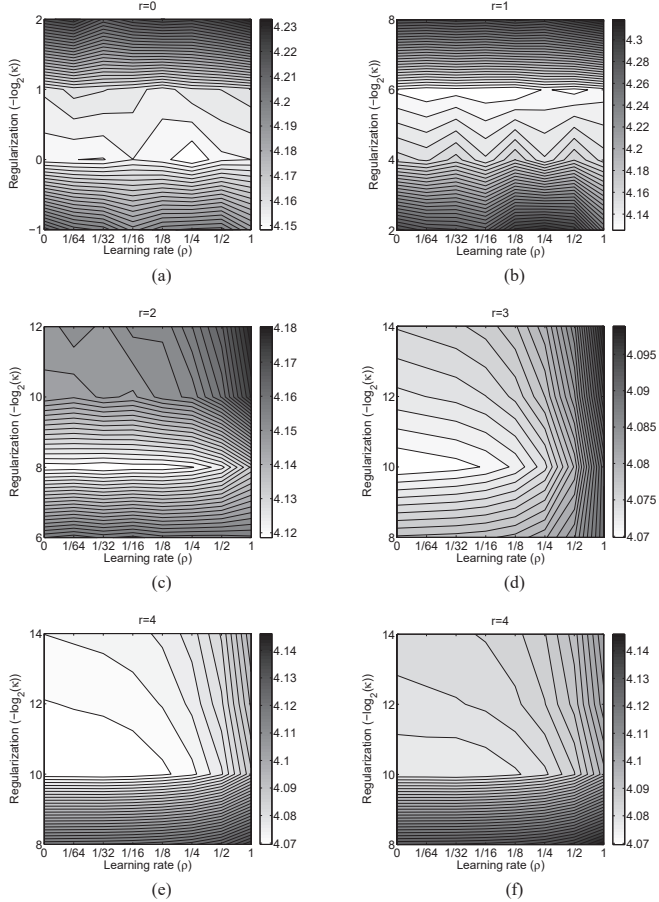


Figure 4.6: Validation surfaces [(a)-(e)] and test surfaces (f) as a function of forgetting factor (ρ) and regularization (κ). For a fixed (κ, ρ) parameter pair, the surfaces show the best RMSE values optimized in the β similarity parameter. The group structure (\mathcal{G}) is toroid. The applied neighbor correction was S_1^0 . (a): $r = 0$ (no structure). (b): $r = 1$. (c): $r = 2$. (d): $r = 3$. (e)-(f): $r = 4$, on the same scale.

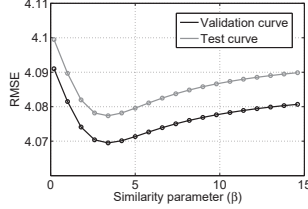


Figure 4.7: Validation and test curves for toroid group structure using the optimal neighborhood size $r = 4$, regularization weight $\kappa = \frac{1}{2^{10}}$, forgetting factor $\rho = \frac{1}{2^5}$, mini-batch size $R = 8$, and similarity parameter $\beta = 3.4$. The applied neighbor correction was S_1^0 .

Table 4.3: Performance (RMSE) of the OSDL prediction using toroid group structure (\mathcal{G}) with different neighbor sizes r ($r = 0$: unstructured case). First-second row: mini-batch size $R = 8$, third-fourth row: $R = 16$. Odd rows: S_1^0 , even rows: S_2^0 neighbor correction. For fixed R , the best performance is highlighted with boldface typesetting.

		$r = 0$	$r = 1$	$r = 2$	$r = 3$	$r = 4$
$R = 8$	S_1^0	4.1594	4.1326	4.1274	4.0792	4.0774
	S_2^0	4.1765	4.1496	4.1374	4.0815	4.0802
$R = 16$	S_1^0	4.1611	4.1321	4.1255	4.0804	4.0777
	S_2^0	4.1797	4.1487	4.1367	4.0826	4.0802

- our method is robust to the selection of correction methods. Similarly to the ρ and R parameters, the neighbor correction parameter can help in fine-tuning the results.
- The introduction of γ_0 in (4.4) with the application of S_1^0 and S_2^0 instead of S_1 and S_2 proved to be advantageous in the neighbor correction phase.
- For the studied CF problem, the S_1^0 neighbor correction method (with $R = 8$) lead to the smallest RMSE value, 4.0774.
- The $R \in \{8, 16\}$ setting yielded us similarly good results. Even with $R = 16$, the RMSE value was 4.0777.

Table 4.4: Performance (RMSE) of the OSDL prediction for different neighbor corrections using toroid group structure (\mathcal{G}). Columns: applied neighbor corrections. Rows: mini-batch size $R = 8$ and 16 . The neighbor size was set to $r = 4$. For fixed R , the best performance is highlighted with boldface typesetting.

	S_1	S_2	S_1^0	S_2^0
$R = 8$	4.0805	4.0844	4.0774	4.0802
$R = 16$	4.0809	4.0843	4.0777	4.0802

Hierarchical Group Structure. In this section we provide results using hierarchical α representation. The group structure \mathcal{G} was chosen to represent a complete binary tree.

In our **third experiment** we study how the number of levels (l) of the tree affects the results. To this end, we set the number of levels to $l = 3, 4, 5$, and 6 . Since d_α , the dimension of the hidden representation α , equals to $2^l - 1$, these l values give rise to dimensions $d_\alpha = 7, 15, 31$, and 63 . Validation and test surfaces are provided in Fig. 4.8(a)-(c) and (e)-(f), respectively. The surfaces show for each (κ, ρ) pair, the minimum RMSE values taken in the similarity parameter β . For the best (κ, ρ) parameter pair, the dependence of RMSE on β is presented in Fig. 4.8(d). In this illustration we used S_1^0 neighbor correction, and the mini-batch size was set to $R = 8$. Our results are summarized below. We note that we obtained similar results with mini-batch size $R = 16$.

- For fixed number of levels l , similarly to the toroid group structure (where the size r of the neighborhood was fixed),
 - validation and test surfaces are very similar, see Fig. 4.8(b)-(c). Validation and test curves as a function of β behave alike, see Fig. 4.8(d).
 - the precision of the estimation depends mostly on the regularization parameter κ ; forgetting factor ρ enables fine-tuning.
- The obtained RMSE values are summarized in Table 4.5 for S_1^0 and S_2^0 neighbor corrections. According to the table, the quality of estimation is about the same for mini-batch size $R = 8$ and $R = 16$; the $R = 8$ based estimation seems somewhat more precise. Considering the neighbor correction schemes S_1^0 and S_2^0 , S_1^0 provided better predictions.

Table 4.5: Performance (RMSE) of the OSDL prediction for different number of levels (l) using binary tree structure (\mathcal{G}). First-second row: mini-batch size $R = 8$, third-fourth row: $R = 16$. Odd rows: S_1^0 , even rows: S_2^0 neighbor correction. For fixed R , the best performance is highlighted with boldface typesetting.

		$l = 3$	$l = 4$	$l = 5$	$l = 6$
$R = 8$	S_1^0	4.1572	4.1220	4.1241	4.1374
	S_2^0	4.1669	4.1285	4.1298	4.1362
$R = 16$	S_1^0	4.1578	4.1261	4.1249	4.1373
	S_2^0	4.1638	4.1332	4.1303	4.1383

- As a function of the number of levels, we got the best result for $l = 4$, RMSE = 4.1220; RMSE values decrease until $l = 4$ and then increase for $l > 4$.
- Our best obtained RMSE value is 4.1220; it was achieved for dimension only $d_\alpha = 15$. We note that this small dimensional, hierarchical group structure based result is also better than that of [182] with RMSE = 4.1229, which makes use of unstructured dictionaries with $d_\alpha = 100$. The result is also competitive with the RMSE = 4.1123 value of [184].

In our **fourth experiment** we investigate how the different neighbor corrections (S_1 , S_2 , S_1^0 , S_2^0) affect the precision of the estimations. We fixed the number of levels to $l = 4$, since it proved to be the optimal choice in our previous experiment. Our results are summarized in Table 4.6. We found that

- the estimation is robust to the choice of neighbor corrections,
- it is worth including weight γ_0 [see (4.4)] to improve the precision of prediction, that is, to apply correction S_1^0 and S_2^0 instead of S_1 and S_2 , respectively.
- the studied $R \in \{8, 16\}$ mini-batch sizes provided similarly good results.
- for the studied CF problem the best RMSE value was achieved using S_1^0 neighbor correction and mini-batch size $R = 8$.

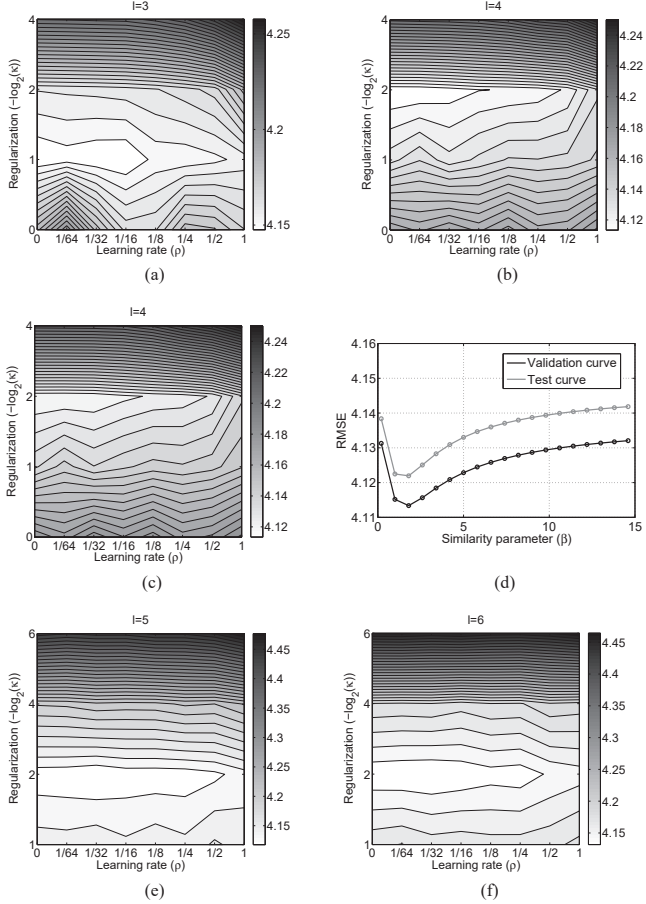


Figure 4.8: Validation surfaces [(a)-(b), (e)-(f)] and test surfaces (c) as a function of forgetting factor (ρ) and regularization (κ) . (d): validation and test curve using the optimal number of levels $l = 4$, regularization weight $\kappa = \frac{1}{2^2}$, forgetting factor $\rho = 0$, mini-batch size $R = 8$, similarity parameter $\beta = 1.8$. Group structure (G): complete binary tree. Neighbor correction: S_0^0 . (a)-(c),(e)-(f): for fixed (κ, ρ) parameter pair, the surfaces show the best RMSE values optimized in the β similarity parameter. (a): $l = 3$. (b)-(c): $l = 4$, on the same scale. (e): $l = 5$. (f): $l = 6$.

Table 4.6: Performance (RMSE) of the OSDL prediction for different neighbor corrections using binary tree structure (\mathcal{G}). Rows: mini-batch size $R = 8$ and 16. Columns: neighbor corrections. Neighbor size: $r = 4$. For fixed R , the best performance is highlighted with boldface typesetting.

	S_1	S_2	S_1^0	S_2^0
$R = 8$	4.1255	4.1338	4.1220	4.1285
$R = 16$	4.1296	4.1378	4.1261	4.1332

Chapter 5

Numerical Experiments – Independent Subspace Based Dictionary Learning

In this chapter we illustrate the efficiency of the proposed IPA estimation methods (Chapter 3). Test databases are described in Section 5.1. To evaluate the solutions, we use the performance measure given in Section 5.2. Our numerical results are presented in Section 5.3.

5.1 Test Datasets

We conducted experiments using the following datasets to assess the efficiency, robustness and limits of our methods:

ABC, 3D-geom: In the *ABC* database, the distribution of the hidden sources \mathbf{e}^m were uniform on 2-dimensional images ($d_m = 2$) of the English alphabet. The number of components can be $M = 26$. For illustration, see Fig. 5.1(b).

In the *3D-geom* test \mathbf{e}^m s were random variables uniformly distributed on 3-dimensional geometric forms ($d_m = 3$, $M = 6$), see Fig. 5.1(a).

celebrities, smiley: The *celebrities* and *smiley* test has 2-dimensional source components ($d_m = 2$) generated from cartoons of celebrities ($M = 10$) and 6 basic facial expressions ($M = 6$), respectively.¹ Sources \mathbf{e}^m were generated by

¹ See <http://www.smileyworld.com>.

sampling 2-dimensional coordinates proportional to the corresponding pixel intensities. In other words, 2-dimensional images were considered as density functions. For illustration, see Fig. 5.1(c)-(d).

d-geom, d-spherical: In the *d-geom* dataset \mathbf{e}^m s were random variables uniformly distributed on d_m -dimensional geometric forms. Geometrical forms were chosen as follows. We used: (i) the surface of the unit ball, (ii) the straight lines that connect the opposing corners of the unit cube, (iii) the broken line between $d_m + 1$ points $\mathbf{0} \rightarrow \mathbf{e}_1 \rightarrow \mathbf{e}_1 + \mathbf{e}_2 \rightarrow \dots \rightarrow \mathbf{e}_1 + \dots + \mathbf{e}_{d_m}$ (where \mathbf{e}_i is the i canonical basis vector in \mathbb{R}^{d_m} , i.e., all of its coordinates are zero except the i^{th} , which is 1), and (iv) the skeleton of the unit square. Thus, the number of components M was equal to 4, and the dimension of the components (d_m) can be scaled and different. For illustration, see Fig. 5.1(f).

In the *d-spherical* test hidden sources \mathbf{e}^m were spherical random variables [185]. Since spherical variables assume the form $\mathbf{v} = \rho \mathbf{u}$, where \mathbf{u} is uniformly distributed on the d_m -dimensional unit sphere, and ρ is a non-negative scalar random variable independent of \mathbf{u} , they can be given by means of ρ . We chose 3 pieces of stochastic representations ρ : ρ was uniform on $[0, 1]$, exponential with parameter $\mu = 1$ and lognormal with parameters $\mu = 0$, $\sigma = 1$. For illustration, see Fig. 5.1(g). In this case, the number of component was $M = 3$, and the dimension of the source components (d_m) can be varied.

ikedata: In the *ikedata* test, the hidden $\mathbf{s}_t^m = [s_{t,1}^m, s_{t,2}^m] \in \mathbb{R}^2$ sources realized the ikeda map

$$s_{t+1,1}^m = 1 + \lambda_m [s_{t,1}^m \cos(w_t^m) - s_{t,2}^m \sin(w_t^m)], \quad (5.1)$$

$$s_{t+1,2}^m = \lambda_m [s_{t,1}^m \sin(w_t^m) + s_{t,2}^m \cos(w_t^m)], \quad (5.2)$$

where λ_m is a parameter of the dynamical system and

$$w_t^m = 0.4 - \frac{6}{1 + (s_{t,1}^m)^2 + (s_{t,2}^m)^2}. \quad (5.3)$$

$M = 2$ was chosen with initial points $\mathbf{s}_1^1 = [20; 20]$, $\mathbf{s}_1^2 = [-100; 30]$ and

parameters $\lambda_1 = 0.9994$, $\lambda_2 = 0.998$, see Fig. 5.1(e) for illustration.

all- k -independent: In the *all- k -independent* database, the d_m -dimensional hidden components $\mathbf{v} := \mathbf{e}^m$ were created as follows: coordinates v_i ($i = 1, \dots, k$) were independent uniform random variables on the set $\{0, \dots, k-1\}$, whereas v_{k+1} was set to $\text{mod}(v_1 + \dots + v_k, k)$. In this construction, every k -element subset of $\{v_1, \dots, v_{k+1}\}$ is made of independent variables and $d_m = k + 1$.

Beatles: Our *Beatles* test is a non-i.i.d. example. Here, hidden sources are stereo Beatles songs.² 8 kHz sampled portions of two songs (A Hard Day’s Night, Can’t Buy Me Love) made the hidden s^m s. Thus, the dimension of the components d_m was 2, the number of the components M was 2, and the dimension of the hidden source D was 4.

5.2 Performance Measure, the Amari-index

Below, we present the performance index that was used to measure the quality of the estimations.

First, we focus on the ISA problem. Identification of the ISA model is ambiguous. However, the ambiguities of the model are simple: hidden components can be determined up to permutation of the subspaces and up to invertible linear transformations within the subspaces [153, 186]. Thus, in the ideal case, the product of the estimated ISA demixing matrix $\hat{\mathbf{W}}_{\text{ISA}}$ and the ISA mixing matrix \mathbf{A} , i.e., matrix

$$\mathbf{G} = \hat{\mathbf{W}}_{\text{ISA}} \mathbf{A} \quad (5.4)$$

is a block-permutation matrix (also called block-scaling matrix [110]). This property can also be measured for source components with different dimensions by a simple extension [164] of the Amari-index [187], that we present below. Namely, assume that we have a weight matrix $\mathbf{V} \in \mathbb{R}^{M \times M}$ made of positive matrix elements. Loosely speaking, we shrink the $d_i \times d_j$ blocks of matrix \mathbf{G} according to the weights of matrix \mathbf{V} and apply the traditional Amari-index for the matrix we

²See <http://rock.mididb.com/beatles/>.

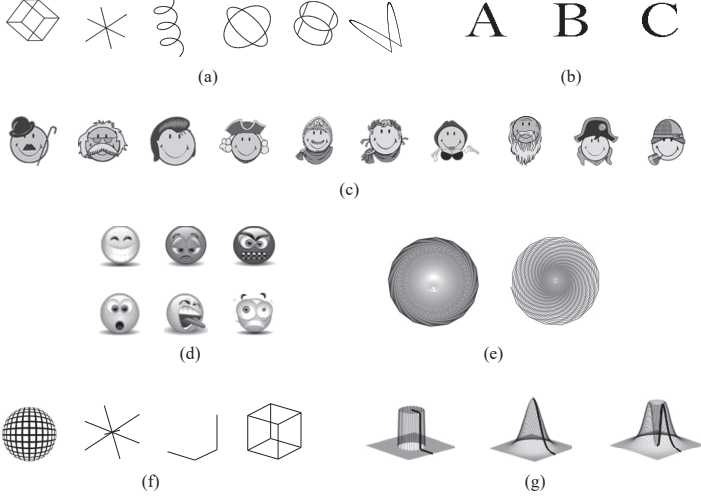


Figure 5.1: Illustration of the *3D-geom* (a), *ABC* (b), *celebrities* (c), *smiley* (d), *ikeda* (e), *d-geom* (f) and *d-spherical* (g) datasets.

obtain. Formally, one can (i) assume without loss of generality that the component dimensions and their estimations are ordered in increasing order ($d_1 \leq \dots \leq d_M$, $\hat{d}_1 \leq \dots \leq \hat{d}_M$), (ii) decompose \mathbf{G} into $d_i \times d_j$ blocks ($\mathbf{G} = [\mathbf{G}^{ij}]_{i,j=1,\dots,M}$) and define g^{ij} as the sum of the absolute values of the elements of the matrix $\mathbf{G}^{ij} \in \mathbb{R}^{d_i \times d_j}$, weighted with V_{ij} :

$$g^{ij} = V_{ij} \sum_{k=1}^{d_i} \sum_{l=1}^{d_j} |(\mathbf{G}^{ij})_{k,l}|. \quad (5.5)$$

Then the Amari-index with parameters \mathbf{V} can be adapted to the ISA task of possibly different component dimensions as follows

$$r_{\mathbf{V}}(\mathbf{G}) := \frac{1}{2M(M-1)} \left[\sum_{i=1}^M \left(\frac{\sum_{j=1}^M g^{ij}}{\max_j g^{ij}} - 1 \right) + \sum_{j=1}^M \left(\frac{\sum_{i=1}^M g^{ij}}{\max_i g^{ij}} - 1 \right) \right]. \quad (5.6)$$

One can see that $0 \leq r_{\mathbf{V}}(\mathbf{G}) \leq 1$ for any matrix \mathbf{G} , and $r_{\mathbf{V}}(\mathbf{G}) = 0$ if and only if \mathbf{G} is block-permutation matrix with $d_i \times d_j$ sized blocks. $r_{\mathbf{V}}(\mathbf{G}) = 1$ is in the worst case, i.e., when all the g^{ij} elements are equal. Let us note that this novel measure (5.6) is invariant, e.g., for multiplication with a positive constant: $r_{c\mathbf{V}} = r_{\mathbf{V}}$ ($\forall c > 0$). Weight matrix \mathbf{V} can be uniform ($V_{ij} = 1$), or one can use weighing according to the size of the subspaces: $V_{ij} = 1/(d_i d_j)$. We will use the shorthand $r(\cdot)$ for the first variant, if not stated otherwise.

Similarly, for the problems presented in Chapter 3, one can estimate the hidden source components only up to the ISA ambiguities. Thus, having the mixing matrix \mathbf{A} at hand, the performance of the estimations can be measured by the block-permutation property of matrix $\mathbf{G} = \hat{\mathbf{W}}_{\text{ISA}} \mathbf{A}$, where $\hat{\mathbf{W}}_{\text{ISA}}$ denotes the estimated demixing matrix of the derived ISA subproblems. In case of the

- complex ISA problem, we measure the block-permutation property of $\mathbf{G} = \hat{\mathbf{W}}_{\text{ISA}} \varphi_M(\mathbf{A})$ using the associated component dimensions over the real domain, i.e., $2 \times d_m$ ($m = 1, \dots, M$).
- BSSD problem, where the mixing is described by a convolution instead of $\mathbf{x} = \mathbf{A}\mathbf{s}$, we chose \mathbf{G} as the linear transformation that optimally approximates the relation $\mathbf{s} \mapsto \hat{\mathbf{s}}$, where $\hat{\mathbf{s}}$ denotes the estimated hidden source.

5.3 Numerical Results

Here, we illustrate the efficiency of the proposed IPA estimation techniques. In Section 5.3.1, Section 5.3.2, Section 5.3.3, Section 5.3.4 and Section 5.3.5 we are dealing with the ARX-IPA, mAR-IPA, complex ISA, fAR-IPA and complete BSSD problem, respectively. Numerical results demonstrating the efficiency of random projection based entropy estimations are given in Section 5.3.6.

In our numerical experiments, the ISA subtask was solved according to the ISA separation theorem [92, 123]: we grouped/clustered the computed ICA components. One may apply different clustering methods—beyond the exhaustive search, which becomes rapidly prohibitive as the dimension of the problem is increasing—e.g.,

Greedy search: We exchange two estimated ICA components belonging to different subspaces, if the exchange decreases the value of the ISA cost as long as

such pairs exist.

Global search: One may apply global permutation search methods of higher computational burden. The cross-entropy solution suggested for the traveling salesman problem (TSP) [188] can, for example, be adapted to our case [189]. In the TSP problem, a permutation of cities is searched for and the objective is to minimize the cost of the travel. We are also searching for a permutation, but now the travel cost is replaced by the ISA cost function.

Spectral clustering: An efficient method with good scaling properties has been put forth in [126, 190] for searching the permutation group for the ISA separation theorem (see, Table 5.1). This approach builds upon the fact that the mutual information between different ISA subspaces e^m is zero due the assumption of independence. The method assumes that coordinates of e^m that fall into the same subspace can be paired by using the mutual information between the coordinates only.

The mutual information of the computed ICA elements can be efficiently estimated, e.g., by the generalized variance [162], the kernel canonical correlation analysis (KCCA) method [191], or the robustness of the estimation against noise can be improved further by applying copula methods [192]. One may carry out the clustering step, e.g., by spectral clustering methods; such a technique is the NCut method [193]. Spectral clustering methods scale well since a single machine can handle a million observations (in our case estimated ICA elements) within several minutes [194].

Finally, it may be worth noting that one can construct examples that do not satisfy the conditions detailed in Table 5.1. The *all-k-independent* construction belongs to this family.

In our experiments the ICA components were estimated by the well-known fastICA algorithm [195]. The performance of our methods are also summarized by notched boxed plots, which show the quartiles (Q_1, Q_2, Q_3), depict the outliers, i.e., those that fall outside of interval $[Q_1 - 1.5(Q_3 - Q_1), Q_3 + 1.5(Q_3 - Q_1)]$ by circles, and whiskers represent the largest and smallest non-outlier data points.

Table 5.1: Approximation that scales well for the permutation search task in the ISA separation theorem.

Construct an undirected graph with nodes corresponding to ICA coordinates and edge weights (similarities) defined by the *pairwise* statistical dependencies, i.e., the mutual information of the estimated ICA elements: $\mathbf{S} = [\hat{I}(\hat{e}_{\text{ICA},i}, \hat{e}_{\text{ICA},j})]_{i,j=1}^D$. Cluster the ICA elements, i.e., the nodes using similarity matrix \mathbf{S} .

5.3.1 ARX-IPA Experiments

Here, we illustrate the efficiency of the proposed ARX-IPA estimation technique (Section 3.1) [152]; results on databases *3D-geom* ($d_m = 3$, $M = 6$, $D_s = 3 \times 6 = 18$), *ABC* ($d_m = 2$, $M = 10$, $D_s = 2 \times 10 = 20$) and *celebrities* ($d_m = 2$, $M = 10$, $D_s = 2 \times 10 = 20$) are provided.³ For each individual parameter, the performance of 20 random runs were averaged. Our parameters are: T , the sample number of observations \mathbf{x}_t , L_s , the order of dynamics of the AR part, L_u , the temporal memory of the effect of the control applied, δ_u , the upper limit of the magnitude of the control ($U := \{\mathbf{u} : \max_i |u_i| \leq \delta_u\}$), and λ , parameter of the stable $\mathbf{F}[z]$. ‘Random run’ means random choice of quantities $\mathbf{F}[z]$, \mathbf{B}_j s, \mathbf{A} and \mathbf{e} . In each simulation \mathbf{A} was a random orthogonal matrix⁴, sample number T varied between 1,000 and 100,000, we optimized J_{par} and J_{noise} on intervals $[1, T/2]$ and $[T/2 + 1, T]$, respectively (see footnote 3), the dimension of the control was equal to the dimension of \mathbf{s} ($D_u = D_s$), the ISA task was solved by using the JFD (joint f-decorrelation; generalized variance dependence, greedy permutation search) method [162], the elements of matrices \mathbf{B}_j were generated independently from standard normal distributions, and stable the $\mathbf{F}[z]$ was generated as follows

$$\mathbf{F}[z] = \prod_{i=0}^{L_s-1} (\mathbf{I} - \lambda \mathbf{O}_i z) \quad (|\lambda| < 1, \lambda \in \mathbb{R}), \quad (5.7)$$

³We note that the InfoMax objectives J_{par} and J_{noise} look forward only by one-step, so the method is greedy. The objective could be extended to include long-term cumulated contributions, but the solution is not yet known for this task. According to experiences, estimation of noise \mathbf{e} can proceed by using J_{par} first for a some iterations and then use J_{noise} to compute the control values [151].

⁴In our studied ISA based problems, one can assume without loss of generality that the \mathbf{A} mixing matrix belongs to the orthogonal family, this corresponds to a simple normalizing assumption.

where matrices $\mathbf{O}_i \in \mathbb{R}^{D_s \times D_s}$ were random orthogonal ($\mathbf{O}_i \in \mathcal{O}^{D_s}$).

We sum up our experiences about the ARX-IPA method here:

1. Dependence on δ_u : We studied the effect of the magnitude of control (δ_u) on the precision of the estimation for ‘small’ L_s, L_u ($L_s, L_u \leq 3$) values and for $\lambda = 0.95$. We found that for a range of not too large control values δ_u the estimation is precise (Fig. 5.2(a)) and the error follows a power law in the number of samples: $r(T) \propto T^{-c}$ ($c > 0$) is a straight line on log-log scale. Similar results were found for all three databases in all experiments (Fig. 5.2(b)). Figure 5.3 illustrates the results of the estimations. In the rest of the studies we fixed the maximum of the control magnitude to $\delta_u = 0.2$ and show the results of the *3D-geom* database.
2. Dependence on L_u : Increasing the temporal memory of the effect of the control applied ($L_u = 3, 5, 10, 20, 50$) precise estimation was found even for $L_u = 50$. The estimation errors are shown in Fig. 5.4(a).
3. Dependencies on L_s and λ : We found that the order of the dynamics of the AR process (L_s) can be increased provided that λ in Eq. (5.7) is decreased: For $L_u = 1$ and for $L_s = 5, 10, 20, 50$, the estimation is precise up to values approximately equal to $\lambda = 0.85 - 0.9, 0.65 - 0.7, 0.45 - 0.5, 0.25 - 0.3$, respectively. Results are depicted in Fig. 5.4(b).

For further illustration concerning the ARMAX-IPA and PNL ARX-IPA models, see [155, 156].

5.3.2 mAR-IPA Experiments

Here, we illustrate the efficiency of the proposed mAR-IPA estimation technique (Section 3.2) [157, 158]; results on databases *ABC* ($d_m = 2, M = 3, D = 2 \times 3 = 6$), *3D-geom* ($d_m = 3, M = 2, D = 3 \times 2 = 6$) and *Beatles* ($d_m = 2, M = 2, D = 2 \times 2 = 4$) are provided. For each individual parameter, the performance of 10 random runs ($\mathbf{A}, \mathbf{F}[z], \mathbf{e}$) were averaged. Our parameters are: T , the sample number of observations \mathbf{y}_t , L_s , the order of the AR process, p , the probability of missing observation in \mathcal{M}_t ($x_{t,i}$ s, the coordinates of process \mathbf{x}_t , were not observed

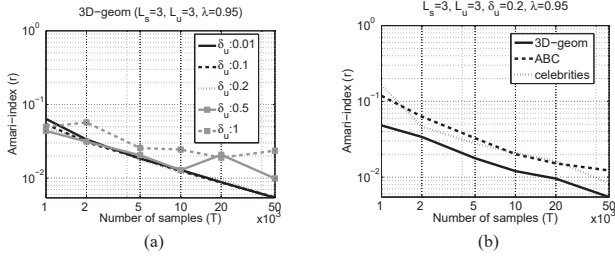


Figure 5.2: ARX-IPA problem, estimation error (Amari-index) as a function of sample number on log-log scale for different control magnitudes (a), and databases (b).

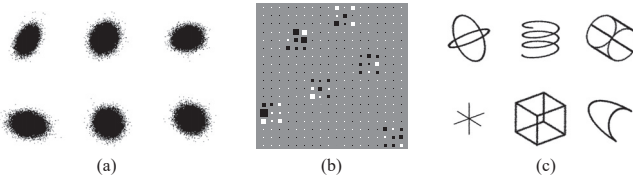


Figure 5.3: ARX-IPA problem, illustration for the 3D-geom database ($L_s = L_u = 3$, $\delta_u = 0.2$, $\lambda = 0.95$, $T = 50,000$), for an estimation with average estimation error ($100 \times \text{Amari-index} = 0.55\%$). (a): observed signal \mathbf{x}_t . (b) Hinton-diagram of G : the product of the estimated demixing matrix and the mixing matrix of the derived ISA task (= approximately block-permutation matrix with 3×3 blocks). (c): estimated components—recovered up to the ISA ambiguities.

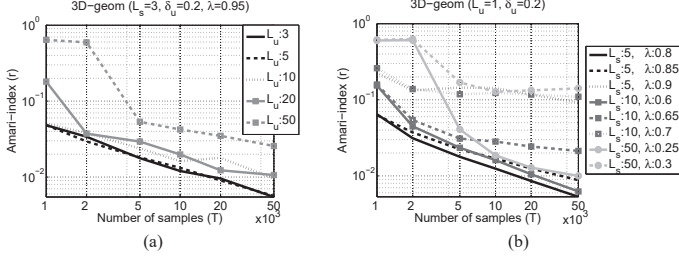


Figure 5.4: ARX-IPA problem, estimation error (Amari-index) as a function of (a) temporal memory of control L_u , and (b) order of the AR process L_s .

with probability p , independently), and λ , the (contraction) parameter of the stable polynomial matrix $\mathbf{F}[z]$. It is expected that if the roots of $\mathbf{F}[z]$ are close to the unit circle then our estimation will deteriorate. We investigated this by generating the polynomial matrix $\mathbf{F}[z]$ as

$$\mathbf{F}[z] = \prod_{i=0}^{L_s-1} (\mathbf{I} - \lambda \mathbf{O}_i z) \quad (|\lambda| < 1, \lambda \in \mathbb{R}), \quad (5.8)$$

where matrices $\mathbf{O}_i \in \mathbb{R}^{D \times D}$ were random orthogonal ($\mathbf{O}_i \in \mathcal{O}^D$) and the $\lambda \rightarrow 1$ limit was studied. Mixing matrix \mathbf{A} was a random orthogonal matrix. AR fit subject to missing observations was accomplished by means of (i) the maximum likelihood (ML) principle [196], (ii) the subspace technique [197], and (iii) in a Bayesian framework using normal-inverted Wishart (shortly *NIW*) conjugate prior and filling in the next missing data using the maximum-a-posteriori estimation of the parameters [198]. The dependency of the estimated ICA elements elements was estimated by means of the KCCA method [191]. The performance of the method is summarized by notched boxed plots.

The L_s order of the AR process was 1 and 2 for the *ABC* and the *3D-geom* tasks, respectively, contraction parameter λ was varied between values 0.1 and 0.99, the probability of missing observations took different values ($p = 0.01, 0.1, 0.15, 0.2$), and sample number T was set to 1,000, 2,000, and 5,000. According to our

experiences, the methods are efficient on both tasks. The most precise method is *ML* followed by the *subspace* method and the *NIW* technique (see Fig. 5.5(a)). Running time of the algorithms is the opposite and the *ML* technique is computation time demanding (see Fig. 5.5(b)). Considering the ratio of missing observations – in the parameter range we studied – the *ML*, the *subspace* and the *NIW* method can handle parameter p up to $0.2 - 0.3$ (see Fig. 5.5(c)-(d)), $p = 0.15 - 0.2$, and $p = 0.1 - 0.15$, respectively. Figure 5.5(c)-(d) demonstrate that the *ML* method works robustly for the contraction parameter λ and provides reasonable estimations for values around 1. Figure 5.5(e)-(j) illustrate the *ML* component estimations for different p values.

Because of the high computation demands of the *ML* technique, the performances of the *subspace* and *NIW* methods were studied on the *Beatles* test. According to the Schwarz’s Bayesian criterion we used the crude $L_s = 10$ AR estimation. Results for sample number $T = 30,000$ are summarized in Fig. 5.6. According to the figure, the methods give reasonable estimations up to $p = 0.1 - 0.15$. In accord with our previous experiences, the *subspace* method is more precise, but it is somewhat slower.

5.3.3 Complex ISA Experiments

Here, we illustrate the efficiency of the presented complex ISA method (Section 3.3) [163]. We provide empirical results on the *d-spherical* dataset ($M = 3$). In our experiments, the $\mathbf{e}^m \in \mathbb{C}^{d_m}$ complex source components were defined by the $2d_m$ -dimensional *d-spherical* construction making use of the φ_v bijection. By the technique described in Section 3.3 the complex ISA problem was mapped to a real valued ISA problem. Then, the KCCA technique [191] was applied to estimate the dependence of the estimated ICA elements. The dimension of the complex components (d_m) were unknown to the algorithm, the clustering of the computed ICA coordinates and the estimation of the component dimensions were accomplished by the NCut [193] spectral clustering technique.

For all parameter values, the average performances upon 10 random initializations of \mathbf{e} and \mathbf{A} were taken. Our parameters included T , the sample number of observations \mathbf{x}_t , and d_m s, the dimensions of the components.⁵ The mixing matrix

⁵In the Amari-index the possible non-equality of the component dimensions (d_m) were also taken

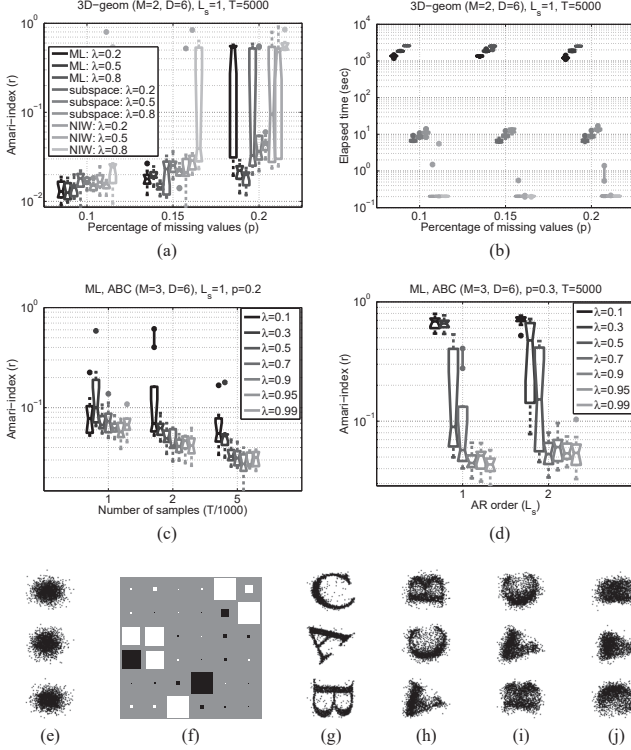


Figure 5.5: mAR-IPA problem, illustration of the estimations on the $3D\text{-geom}$ and ABC datasets. (a), (b): Amari-index and elapsed time, respectively as a function of the probability of missing observation (p) for the $3D\text{-geom}$ dataset on log-log scale and for AR order $L_s = 1$ and sample number $T = 5,000$. (c)-(d): Amari-index for the ML method for $p = 0.2$ and for $p = 0.3$ as a function of the AR order for the ABC test. (e)-(j): illustration of the estimation for the ML method: $L_s = 1$, $T = 5,000$, $\lambda = 0.9$; (e) observation before mapping $\mathcal{M}_t(\mathbf{x})$. (g): estimated components ($\hat{\mathbf{e}}^m$) with average Amari-index for $p = 0.01$. (f): Hinton-diagram of matrix G for (g)—it is approximately a block-permutation matrix with 2×2 blocks. (h)-(j): like (g), but for $p = 0.1$, $p = 0.2$, and $p = 0.3$, respectively.

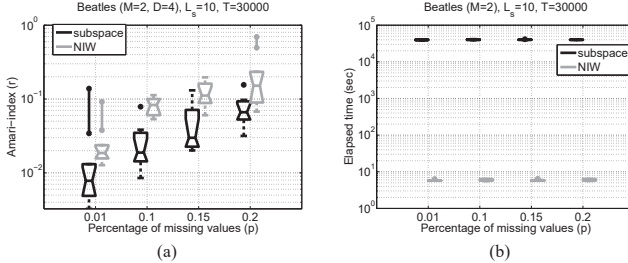


Figure 5.6: mAR-IPA problem, illustration of the *subspace* and the *NIW* methods for the *Beatles* dataset for sample number $T = 30,000$ and AR order $L_s = 10$. (a): Amari-index as a function of the rate of missing observations p on log-log scale, (b): elapsed time.

\mathbf{A} was chosen uniformly from the unitary group \mathcal{U}^D ($D = \sum_{m=1}^M d_m$).⁶ The sample number of observations \mathbf{x}_t changed as $2,000 \leq T \leq 50,000$.

In the first experiment the (complex) dimension of the hidden sources were equal, and varied as $k \times [1; 1; 1]$ where k was chosen from the set $\{2, 3, \dots, 12\}$. We investigated the estimation error as a function of the sample number. Our results for the obtained average Amari-indices are summarized in Fig. 5.7(a). The figure demonstrates that the algorithm was able to estimate the hidden components with high precision. Moreover, as it can be seen the estimation errors are approximately linear as a function of the sample number, that is the Amari-index decreases according to power law $r(T) \propto T^{-c}$ ($c > 0$). The estimated source components are illustrated by Hinton-diagrams, see Fig. 5.7(c). Exact numerical values for the estimation errors can be found in Table 5.2.

In our second experiment the (complex) dimension of the sources could be different and took the values $k \times [1; 1; 2]$, where k was the element of the set $\{2, 3, \dots, 12\}$. The obtained performance values are plotted in Fig. 5.7(b). As it can be seen, (i) the method is able to uncover the hidden source components with high

into account through the $V_{ij} = 1/(2d_i 2d_j)$ construction, see Section 5.2. Here, the ‘ $2d_i$ ’ and ‘ $2d_j$ ’ terms correspond to the associated real valued problem dimensions.

⁶Similarly to the real ISA problem, where the mixing matrix \mathbf{A} can be supposed to be orthogonal, here the unitary property of the mixing matrix \mathbf{A} can be assumed without loss of generality.

Table 5.2: $100 \times$ Amari-index (that is, in percentage) for the complex ISA problem on the ' $k \times [1; 1; 1]$ ' test: average \pm standard deviation. Number of samples: $T = 50,000$.

$k = 2$	$k = 4$	$k = 6$
0.55% ($\pm 0.08\%$)	0.70% ($\pm 0.04\%$)	0.83% ($\pm 0.03\%$)
$k = 8$	$k = 10$	$k = 12$
0.96% ($\pm 0.04\%$)	1.07% ($\pm 0.03\%$)	1.18% ($\pm 0.02\%$)

Table 5.3: $100 \times$ Amari-index (that is, in percentage) for the complex ISA problem on the ' $k \times [1; 1; 2]$ ' test: average \pm standard deviation. Number of samples: $T = 50,000$.

$k = 2$	$k = 4$	$k = 6$
0.56% ($\pm 0.04\%$)	0.82% ($\pm 0.03\%$)	0.97% ($\pm 0.02\%$)
$k = 8$	$k = 10$	$k = 12$
1.13% ($\pm 0.02\%$)	1.24% ($\pm 0.02\%$)	1.37% ($\pm 0.03\%$)

precision and the Amari-indices again follow a power law decay. Hinton-diagram of the estimated sources with average Amari-index are presented in Fig. 5.7(d). Exact numerical values for the Amari-indices are given in Table 5.3.

These results show the efficiency of our complex ISA method.

5.3.4 fAR-IPA Experiments

Now we illustrate the efficiency of the fAR-IPA algorithm [164] presented in Section 3.4. We provide empirical results on the *smiley* ($d_m = 2, M = 6, D = 2 \times 6 = 12$), *d-geom* ($d_1 = 2, d_2 = d_3 = 3, d_4 = 4, M = 4, D = 2 + 3 + 3 + 4 = 12$), and *ikedda* datasets ($d_m = 2, M = 2, D = 2 \times 2 = 4$). For illustration purposes, we chose fAR order $L_s = 1$ and used the recursive Nadaraya-Watson (3.34) for functional AR estimation with the Gaussian kernel. The KCCA technique [191] was applied to estimate the dependence of the computed ICA elements. The clustering was carried out by greedy optimization for tasks when the component dimensions were known (*smiley*, *ikedda* datasets). We also studied the case when these component dimensions were unknown (*d-geom* dataset); in this case we used the NCut [193] spectral technique to cluster the estimated ICA components into ISA subspaces. Mixing

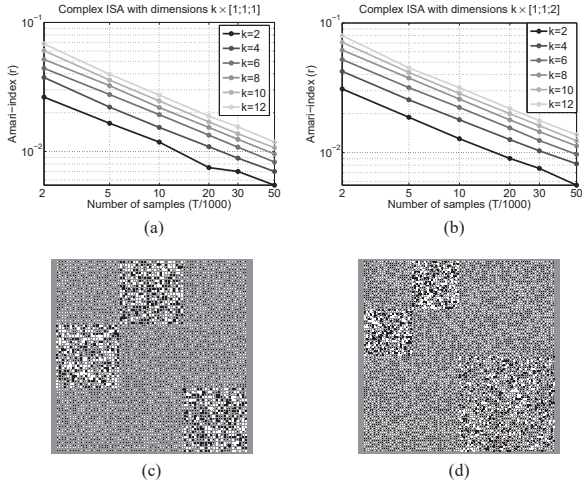


Figure 5.7: Illustration of the complex ISA estimations. (a)-(b): the average Amari-indices are plotted as a function of the sample number on log-log scale. (a): the hidden source dimensions are equal, $k \times [1; 1; 1]$. (b): the hidden source dimension can be different, $k \times [1; 1; 2]$. (c): Hinton-diagram of matrix G with Amari-index closest to the average performance for the ' $k \times [1; 1; 1]$ ' problem with $k = 12$ and sample number $T = 50,000$. The G matrix is approximately block-permutation matrix with $(2 \times 12) \times (2 \times 12)$ sized blocks. (d): the same as (c), but for the different dimensional $k \times [1; 1; 2]$ case with $k = 12$. For exact performance values, see Table 5.2 and Table 5.3.

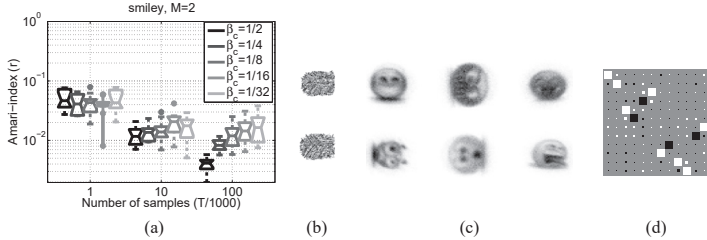


Figure 5.8: fAR-IPA problem, illustration of the estimations on the *smiley* dataset. (a): Amari-index as a function of the sample number, for $M = 2$. (b): observed signal x_t , the first two 2-dimensional projections when $M = 6$. (c): estimated components (\hat{e}^m) with average (closest to the median) Amari-index for $M = 6$, $\beta_c = \frac{1}{32}$, $T = 100,000$. (d): Hinton-diagram of matrix G .

matrix A was random orthogonal. For dataset *smiley* and *d-geom*, f was the composition of a random F matrix with entries distributed uniformly on interval $[0, 1]$ and the noninvertible sine function, $f(u) = \sin(Fu)$. For each individual parameter, the performance of 10 random runs were averaged. Our parameters included T , the sample number of observations x_t , and bandwidth $\beta \in (0, 1/D)$ to study the robustness of the kernel regression approach. β was reparameterized as $\beta = \frac{\beta_c}{D}$ and β_c was chosen from the set $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}\}$. The performance of the method is summarized by notched boxed plots.

For the *smiley* dataset, Fig. 5.8 demonstrates that the algorithm was able to estimate the hidden components with high precision. Fig. 5.8(a) shows the Amari-index as a function of the sample number, for $M = 2$ ($D = 4$). The estimation error is plotted on log scale for different bandwidth parameters. Fig. 5.8(c-d) indicate that the problem with $M = 6$ components ($D = 12$) is still amenable to our method when the sample size is large enough ($T = 100,000$). Fig. 5.8(c) shows the estimated subspaces, and Fig. 5.8(d) presents the Hinton-diagram. It is approximately a block-permutation matrix with 2×2 blocks indicating that the algorithm could successfully estimate the hidden subspaces.

Our experiences concerning the *d-geom* dataset are summarized in Fig. 5.9. In contrast to the previous experiment, here the dimensions of the hidden com-

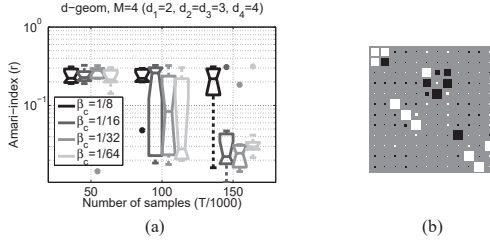


Figure 5.9: fAR-IPA problem, illustration of the estimations on the *d-geom* dataset. (a) Amari-index on log scale as a function of the sample number for different bandwidth parameters on the *d-geom* dataset (with component dimensions: $d_1 = 2$, $d_2 = d_3 = 3$, $d_4 = 4$). (b): Hinton-diagram of G with average (closest to the median) Amari-index for dataset *d-geom*, $\beta_c = \frac{1}{32}$, $T = 150,000$ —it is approximately a block-permutation matrix with one 2×2 , two 3×3 and one 4×4 sized block.

ponents were different and unknown to the algorithm. As it can be seen from Fig. 5.9(a), our method provides precise estimations on this dataset for sample size $T = 100,000 - 150,000$. The Hinton-diagram of matrix G with average (closest to the median) Amari-index is depicted in Fig. 5.9(b). Again, this is close to a block-permutation matrix indicating that the proposed method was able to estimate the hidden subspaces.

We ran experiments on the *iked*a dataset too. Fig. 5.10(a) illustrates that if we simply use a standard autoregressive approximation method (AR-IPA) [159], then we cannot find the proper subspaces. Nevertheless, the Amari-index values of Fig. 5.10(a) show that the functional AR-IPA approach was able to estimate the hidden subspaces for sample number $T \geq 10,000$. The figure also shows that the estimation is precise for a wide range of bandwidth parameters. The Hinton-diagram of matrix G with average (closest to the median) Amari-index is depicted in Fig. 5.10(c). This is a block diagonal matrix, which demonstrates that our method was able to separate the mixed subspaces. The estimated hidden sources (with average Amari-index) are illustrated in Fig. 5.10(d).

Our model (Eq. (3.24)-(3.25)) belongs to the family of state space models. Though the dynamics of the hidden variables s_t is nonlinear, one might wonder

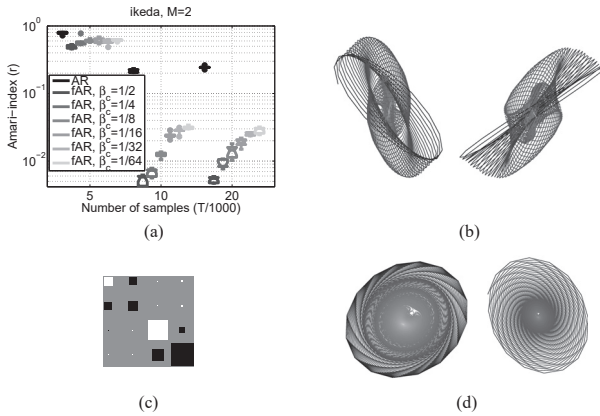


Figure 5.10: Illustration of the estimations on the *ikeda* dataset. (a): Amari-index as a function of the sample number for different bandwidth parameters, for AR-IPA and the proposed fAR-IPA approach. (b): Observation, x_t . (c): Hinton-diagram of G with average (closest to the median) Amari-index. (d): Estimated subspaces using the fAR-IPA method ($\beta_c = \frac{1}{2}$, $T = 20,000$).

whether with a standard linear dynamical system (LDS) based identification method we could identify the parameter \mathbf{A} and the driving noise \mathbf{e}_t . The following experiment demonstrates that this is not the case; while our method is efficiently able to cope with this problem, the LDS based identification leads to very poor results. For this purpose we treated the observations \mathbf{x}_t as if they had been generated by an LDS with unknown parameters. We estimated its parameters with the EM method [199,200], and then using these estimated parameters we applied a Kalman smoother [201] to estimate the hidden dynamical layer \mathbf{s}_t and the driving noise \mathbf{e}_t . After this estimation we post-processed the estimated noise $\hat{\mathbf{e}}_t$ with ISA. We performed these estimations on the *smiley* and *d-geom* datasets. Using 10 independent experiments, the EM-LDS based estimators led to $r = 0.56$ and $r = 0.48$ Amari-indices (minima of the Q_2 medians), respectively. These results are very poor; the EM-LDS based method was not able to identify the noise components. On the contrary, the proposed fAR-IPA method successfully estimated the noise components and provided $r = 0.0041$ and $r = 0.0055$ Amari-indices (Fig. 5.8, Fig. 5.9).

5.3.5 Complete BSSD Experiments

Now we illustrate the efficiency of the complete BSSD method presented in Section 3.5. Results on databases *smiley* ($d_m = 2$, $M = 6$, $D = 2 \times 6 = 12$), *3D-geom* ($d_m = 3$, $M = 4$, $D = 3 \times 4 = 12$) and *Beatles* ($d_m = 2$, $M = 2$, $D = 2 \times 2 = 4$) are provided here. For each individual parameter, the performance of 20 random runs were averaged. Our parameters are: T , the sample number of observations \mathbf{x}_t , L_e , the parameter of the length of the convolution (the length of the convolution is $L_e + 1$), and λ , parameter of the stable $\mathbf{H}[z]$. It is expected that if the roots of $\mathbf{H}[z]$ are close to the unit circle then our estimation will deteriorate, because the stability of $\mathbf{H}[z]$ comes to question. We investigated this by generating the polynomial matrix $\mathbf{H}[z]$ as follows:

$$\mathbf{H}[z] = \left[\prod_{l=0}^{L_e} (\mathbf{I} - \lambda \mathbf{O}_l z) \right] \mathbf{H}_0 \quad (|\lambda| < 1, \lambda \in \mathbb{R}), \quad (5.9)$$

where matrices \mathbf{H}_0 and $\mathbf{O}_i \in \mathbb{R}^{D \times D}$ were random orthogonal ($\mathbf{O}_i \in \mathcal{O}^D$) and the $\lambda \rightarrow 1$ limit was studied. ‘Random run’ means random choice of quantities $\mathbf{H}[z]$

and e. The AR fit to observation \mathbf{x}_t was performed by the method detailed in [202]. To study how the $\mathcal{O}(T^{1/3})$ AR order (see Section 3.5.2) is exploited, the order of the estimated AR process was limited from above by $p_{\max}(T) = 2\lfloor T^{\frac{1}{3} - \frac{1}{1000}} \rfloor$, and we used the Schwarz’s Bayesian criterion to determine the optimal p_{opt} order from the interval $[1, p_{\max}(T)]$. The ISA subtask on the estimated innovation was carried out by the JFD method [162].

First we studied the Amari-index as a function of the sample size. For the *smiley* and *3D-geom* databases the sample number T varied between 1,000 and 20,000. The length of convolution varied as $L_e = 1, 2, 5, 10$. The λ parameter of $\mathbf{H}[z]$ was chosen as 0.4, 0.6, 0.7, 0.8, 0.85, 0.9. Results are shown in Fig. 5.11(a)-(b). The estimation errors indicate that for $L_e = 10$ and about $\lambda = 0.85$ the estimation is still efficient, see Fig. 5.12 for an illustration of the estimated source components. The Amari-indices follow the power law $r(T) \propto T^{-c}$ ($c > 0$). The power law decline is manifested by straight line on log-log scale. The slopes of these straight lines are very close to each other. Numerical values for the estimation errors are given in Table 5.4. The estimated optimal AR orders are provided in Fig. 5.11(c). The figure demonstrates that as $\lambda \rightarrow 1$ the maximal possible order $p_{\max}(T)$ is more and more exploited.

On the *Beatles* database the λ parameter was increased to 0.9, and the sample number T varied between 1,000 and 100,000. Results are presented in Fig. 5.11(d). According to the figure, for $L_e = 1, 2, 5$ the error of estimation drops for sample number $T = 10,000 - 20,000$, and for $L_e = 10$ the ‘power law’ decline of the Amari-index, which was apparent on the *smiley* and the *3D-geom* databases, also appears. Numerical values for the estimation errors are given in Table 5.4. On the *Beatles* test, the maximal possible AR order $p_{\max}(T)$ was fully exploited on the examined parameter domain.

5.3.6 ISA via Random Projections

Now we demonstrate the efficiency of the random projection based entropy estimation presented in Section 3.6 [179] on independent subspace analysis. Results on databases *d-spherical*, *d-geom* and *all-k-independent* are provided here. The experimental studies focused on the following issues:

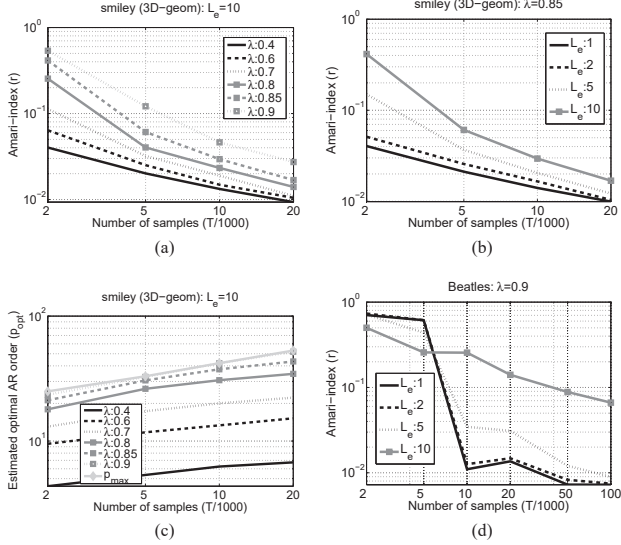


Figure 5.11: Complete BSSD problem, precision of the estimations and the estimated optimal AR orders. The plots are on log-log scale. (a), (b): on the *smiley* (*3D-geom*) database the Amari-index as a function of the sample number for different $\lambda \rightarrow 1$ parameter values of $\mathbf{H}[z]$ and convolution lengths, respectively. In (a): $L_e = 10$, in (b): $\lambda = 0.85$. (c): on the *smiley* (*3D-geom*) database the estimated AR order as a function of the sample number with $L_e = 10$ for different λ values. (d): the same as (b), but for the *Beatles* dataset with $\lambda = 0.9$. For graphical illustration, see Fig. 5.12. For numerical values, see Table 5.4.

Table 5.4: Complete BSSD problem, Amari-index in percentages on the *smiley*, *3D-geom* ($\lambda = 0.85$, $T = 20,000$) and the *Beatles* dataset ($\lambda = 0.9$, $T = 100,000$) for different convolution lengths: mean \pm standard deviation. For other sample numbers, see Fig. 5.11.

	$L_e = 1$	$L_e = 2$	$L_e = 5$	$L_e = 10$
smiley	0.99% ($\pm 0.11\%$)	1.04% ($\pm 0.09\%$)	1.22% ($\pm 0.15\%$)	1.69% ($\pm 0.26\%$)
3D-geom	0.42% ($\pm 0.06\%$)	0.54% ($\pm 0.05\%$)	0.88% ($\pm 0.14\%$)	1.15% ($\pm 0.24\%$)
Beatles	0.72% ($\pm 0.12\%$)	0.75% ($\pm 0.11\%$)	0.90% ($\pm 0.23\%$)	6.64% ($\pm 7.49\%$)

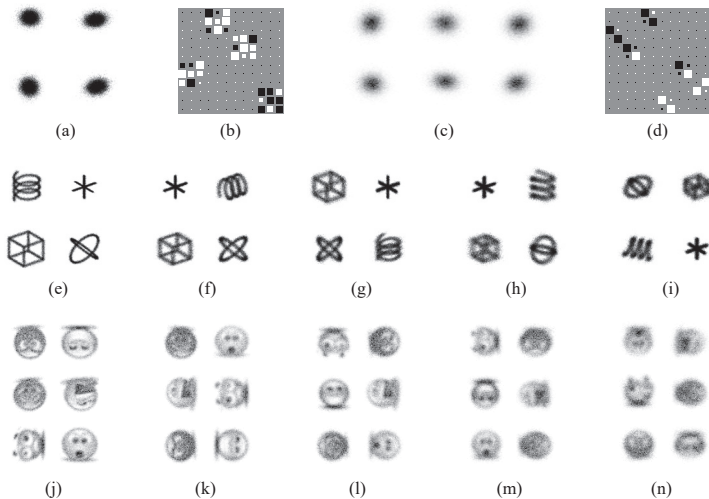


Figure 5.12: Complete BSSD problem, illustration of the estimations on the *3D-geom* [(a),(b),(e)-(i)] and *smiley* [(c),(d),(j)-(n)] datasets. Number of samples: $T = 20,000$. Length of the convolution: $L_e = 10$. In the first row: $\lambda = 0.4$. (a), (c): observed convolved signal x_t . (b), (d): Hinton-diagram of G , ideally a block-permutation matrix with 2×2 and 3×3 sized blocks, respectively. (e)-(i), (j)-(n): estimated components \hat{e}^m , recovered up to the ISA ambiguities from left to right for $\lambda = 0.4, 0.6, 0.7, 0.8, 0.85$. All the plotted estimations have average Amari-indices, see Fig. 5.11(a).

1. What dimensional reduction can be achieved in the entropy estimation of the ISA problem by means of random projections?
2. What speed-up can be gained with the RP dimension reduction?
3. What are the advantages of our RP based approach in global optimization?

In our experiments the number of components was minimal ($M = 2$). For each individual parameter, the performance of 50 random runs were averaged. Our parameters included T , the sample number of observations \mathbf{x}_t and d , the dimension of the components ($d = d_1 = d_2$). We also studied different estimations of the ISA cost function: we used the RADICAL (robust, accurate, direct ICA algorithm) procedure⁷ [203] and the NN method [180] for entropy estimation and KCCA [204] for mutual information estimation. The reduced dimension d' in RP and the optimization method (greedy, global (CE), NCut [126]) of the ISA cost were also varied in different tests. Random run means random choice of quantities \mathbf{A} and \mathbf{e} . The size of the randomly projected groups was set to $|I_n| = 2,000$, except for the case $d = 50$, when it was 5,000. RP was realized by the *database-friendly projection* technique, i.e., the $r_{n,ij}$ coordinates of \mathbf{R}_n were drawn independently from distribution $P(r_{n,ij} = \pm 1) = 1/2$, but more general constructions could also be used [175, 176].

In the first study we were interested in the limits of the RP dimension reduction. We increased dimension d of the subspaces for the *d-spherical* and the *d-geom* databases ($d = 2, 10, 20, 50$) and studied the extreme case, the RP dimension d' was set to 1. Results are summarized in Fig. 5.13(a)-(b) with quartiles (Q_1, Q_2, Q_3). We found that the estimation error decreases with sample number according to a power law [$r(T) \propto T^{-c}$ ($c > 0$)] and the estimation works up to about $d = 50$. For the $d = 50$ case we present notched boxed plots (Fig. 5.13(c)). According to the figure, the error of estimation drops for sample number $T = 100,000$ for both types of datasets: for databases *50-geom* and *50-spherical*, respectively, we have 5 and 9 outliers from 50 random runs and thus with probability 90% and 82%, the estimation is accurate. As for question two, we compared the efficiency (Q_1, Q_2, Q_3) of our

⁷We chose RADICAL, because it is consistent, asymptotically efficient, converges rapidly, and it is computationally efficient. By RADICAL, we mean the spacing based entropy estimation part of the algorithm.

method for $d = 20$ with the NN methods by RP-ing into $d' = 1$ and $d' = 5$ dimensions. Results are shown in Fig. 5.13(e)-(f).⁸ The figure demonstrates that for database *20-geom* performances are similar, but for database *20-spherical* our method has smaller standard deviation for $T = 20,000$. At the same time our method offers 8 to 30 times speed-up at $T = 100,000$ for *serial implementations*. Figure 5.14 presents the components estimated by our method for dimensions $d = 2$ and $d = 50$, respectively. With regard to our third question, the ISA problem can often be solved by grouping the estimated ICA coordinates based on their mutual information. However, this method, as illustrated by (Q_1, Q_2, Q_3) in Fig. 5.13(d), does not work for our *all-4-independent* database. Inserting the RP based technique into global optimization procedure, we get accurate estimation for this case, too. CE optimization was used here. Results are presented in Fig. 5.13(d).

⁸We note that for $d = 20$ and without dimension reduction the NN methods are very slow for the ISA tasks.

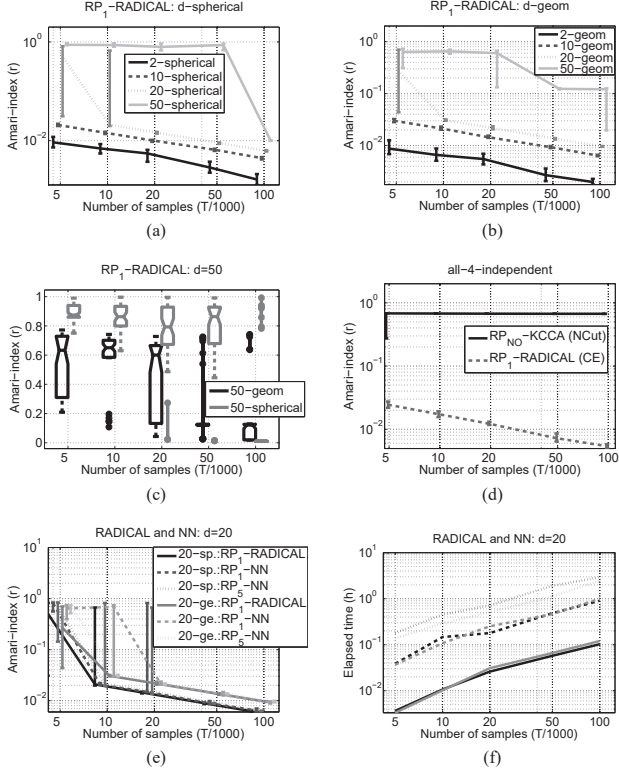


Figure 5.13: Performance of the RP method in ISA. Notations: ‘ RP_d ’ - method of cost estimation (method of optimization if not greedy). (a), (b): accuracy of the estimation versus the number of samples for the d -spherical and the d -geom databases on log-log scale. (c): notched boxed plots for $d = 50$, (d): Performance comparison on the *all-4-independent* database between the RP method using global optimization and the NCut based grouping of the estimated ICA coordinates using the pairwise mutual information graph (on log-log scale). (e)-(f): Accuracy and computation time comparisons with the NN based method for the *20-spherical* and the *20-geom* databases (on log-log scale).

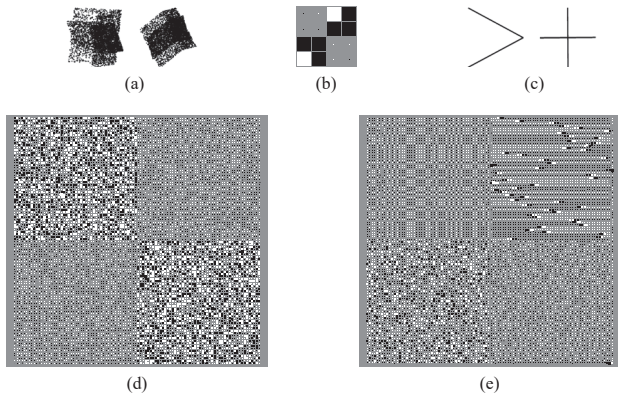


Figure 5.14: RP based ISA, estimated components and Hinton-diagrams. Number of samples: $T = 100,000$. Databases *2-geom*: (a)-(c), *50-spherical*: (d), *50-geom*: (e). (a): observed signals \mathbf{x}_t , (b): Hinton-diagram of \mathbf{G} : the product of the mixing matrix of the ISA task and the estimated demixing matrix is approximately a block-permutation matrix with 2×2 sized blocks, (c): estimated components $\hat{\mathbf{e}}^m$, recovered up to the ISA ambiguities, (d)-(e): Hinton-diagrams of the *50-spherical* and the *50-geom* tests, respectively. Hinton-diagrams have average Amari-indices: for (b) 0.2%, for (d) 1%, for (e) 12%.

Chapter 6

Conclusions

In this thesis we addressed the dictionary learning problem in case of two different assumptions on the hidden sources: (i) group sparsity and (ii) independent subspaces (ISA, independent subspace analysis).

In the former case, we proposed a new dictionary learning method, which (i) is online, (ii) enables overlapping group structures on the hidden representation/dictionary, (iii) applies non-convex, sparsity inducing regularization, and (iv) can handle the partially observable case, too. We reduced the formulated online group-structured dictionary learning (OSDL) problem to convex subtasks, and using a block-coordinate descent approach and a variational method we derived online update rules for the statistics of the cost of the dictionary. The efficiency of our algorithm was demonstrated by several numerical experiments. We have shown that in the inpainting problem of natural images the proposed OSDL method can perform better than the traditional sparse methods. We have shown that our approach can be used for the online structured NMF problem, too, and it is able to hierarchically organize the elements of the dictionary. We have also dealt with collaborative filtering (CF) based recommender systems. Our extensive numerical experiments showed that structured dictionaries have several advantages over the state-of-the-art CF methods: more precise estimation can be obtained, and smaller dimensional feature representation can be sufficient by applying group-structured dictionaries. Moreover, the estimation behaves robustly as a function of the OSDL parameters and the applied group structure.

We derived novel kernel based function approximation techniques and kernel – sparsity equivalences. In particular, we generalized a variant of sparse coding scheme to reproducing kernel Hilbert spaces (RKHS) with component-wise, ϵ -sparse properties and proved that the obtained problem can be transformed to a generalized family of support vector machine (SVM) problem. We also showed that SVMs can be embedded into multilayer perceptrons (MLP) and for the obtained multilayer perceptron architecture the backpropagation procedure of MLPs can be generalized.

We extended the ISA problem to several domains. Our work was motivated by a central result, a 10-year-old unresolved hypothesis of the ICA (independent component analysis) research, the ISA separation principle. This principle (i) enables one to solve the ISA problem via traditional ICA up to permutation, (ii) has been rigorously proven for certain distribution types recently (sufficient conditions are now known for the principle), (iii) forms the basis of the state-of-the-art ISA solvers, (iv) makes it possible to estimate the unknown number and the dimensions of the sources efficiently.

We generalized the ISA problem to numerous new directions including the controlled, the partially observed, the complex valued and the nonparametric case. We derived separation principle based solution techniques for the formulated problems. This approach makes it possible to (i) apply state-of-the-art algorithms for the obtained subproblems (ICA, spectral clustering, D-optimal identification, kernel regression, etc.) and (ii) tackle the case of unknown source component dimensions efficiently. We extended the Amari-index performance measure to different dimensional components. Our extensive numerical illustrations demonstrated the robustness and attractive scaling properties of the approach. The novel models may also lead to a new generation of control assisted data mining applications, interaction paradigms, biomedical, econometric and financial prediction approaches.

Appendix A

Proofs

A.1 Online Group-Structured Dictionary Learning

In this section we focus on the OSDL problem. We will derive the update equations for the statistics describing the minimum point of \hat{f}_t (Section A.1.2). During the derivation we will need an auxiliary lemma concerning the behavior of certain matrix series. We will introduce this lemma in Section A.1.1.

A.1.1 The Forgetting Factor in Matrix Recursions

Let $\mathbf{N}_t \in \mathbb{R}^{L_1 \times L_2}$ ($t = 1, 2, \dots$) be a given matrix series, and let $\gamma_t = \left(1 - \frac{1}{t}\right)^\rho$, $\rho \geq 0$. Define the following matrix series with the help of these quantities:

$$\mathbf{M}_t = \gamma_t \mathbf{M}_{t-1} + \mathbf{N}_t \in \mathbb{R}^{L_1 \times L_2} \quad (t = 1, 2, \dots), \quad (\text{A.1})$$

$$\mathbf{M}'_t = \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \mathbf{N}_i \in \mathbb{R}^{L_1 \times L_2} \quad (t = 1, 2, \dots). \quad (\text{A.2})$$

Lemma 1. *If $\rho = 0$, then $\mathbf{M}_t = \mathbf{M}_0 + \mathbf{M}'_t$ ($\forall t \geq 1$). When $\rho > 0$, then $\mathbf{M}_t = \mathbf{M}'_t$ ($\forall t \geq 1$).*

Proof.

1. Case $\rho = 0$: Since $\gamma_t = 1$ ($\forall t \geq 1$), thus $\mathbf{M}_t = \mathbf{M}_0 + \sum_{i=1}^t \mathbf{N}_i$. We also have that $\left(\frac{i}{t}\right)^0 = 1$ ($\forall i \geq 1$), and therefore $\mathbf{M}'_t = \sum_{i=1}^t \mathbf{N}_i$, which completes the proof.

2. Case $\rho > 0$: The proof proceeds by induction.

- $t = 1$: In this case $\gamma_1 = 0$, $\mathbf{M}_1 = 0 \times \mathbf{M}_0 + \mathbf{N}_1 = \mathbf{N}_1$ and $\mathbf{M}'_1 = \mathbf{N}_1$, which proves that $\mathbf{M}_1 = \mathbf{M}'_1$.
- $t > 1$: Using the definitions of \mathbf{M}_t and \mathbf{M}'_t , and exploiting the fact that $\mathbf{M}_{t-1} = \mathbf{M}'_{t-1}$ by induction, after some calculation we have that:

$$\mathbf{M}_t = \gamma_t \mathbf{M}_{t-1} + \mathbf{N}_t = \left(1 - \frac{1}{t}\right)^\rho \left[\sum_{i=1}^{t-1} \left(\frac{i}{t-1}\right)^\rho \mathbf{N}_i \right] + \mathbf{N}_t \quad (\text{A.3})$$

$$= \left(\frac{t-1}{t}\right)^\rho \left[\sum_{i=1}^{t-1} \left(\frac{i}{t-1}\right)^\rho \mathbf{N}_i \right] + \left(\frac{t}{t}\right)^\rho \mathbf{N}_t \quad (\text{A.4})$$

$$= \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \mathbf{N}_i = \mathbf{M}'_t. \quad (\text{A.5})$$

□

A.1.2 Online Update Equations for the Minimum Point of \hat{f}_t

Our goals are (i) to find the minimum of

$$\hat{f}_t(\mathbf{D}) = \frac{1}{\sum_{j=1}^t (j/t)^\rho} \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \left[\frac{1}{2} \|\mathbf{x}_{O_i} - \mathbf{D}_{O_i} \boldsymbol{\alpha}_i\|_2^2 + \kappa \Omega(\boldsymbol{\alpha}_i) \right] \quad (\text{A.6})$$

in \mathbf{d}_j while the other column vectors of \mathbf{D} (\mathbf{d}_i ($i \neq j$)) are being fixed, and (ii) to derive online update rules for the statistics of \hat{f}_t describing this minimum point. \hat{f}_t is quadratic in \mathbf{d}_j , hence in order to find its minimum, we simply have to solve the following equation:

$$\frac{\partial \hat{f}_t}{\partial \mathbf{d}_j}(\mathbf{u}_j) = 0, \quad (\text{A.7})$$

where \mathbf{u}_j denotes the optimal solution. We can treat the Ω , and the $\frac{1}{\sum_{j=1}^t (j/t)^\rho}$ terms in (A.6) as constants, since they do not depend on \mathbf{d}_j . Let \mathbf{D}_{-j} denote the slightly modified version of matrix \mathbf{D} ; its j^{th} column is set to zero. Similarly, let $\boldsymbol{\alpha}_{i,-j}$

denote the vector α_i where its j^{th} coordinate is set to zero. Now, we have that

$$\mathbf{0} = \frac{\partial \hat{f}_t}{\partial \mathbf{d}_j} = \frac{\partial}{\partial \mathbf{d}_j} \left[\sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \|\Delta_i(\mathbf{x}_i - \mathbf{D}\alpha_i)\|_2^2 \right] \quad (\text{A.8})$$

$$= \frac{\partial}{\partial \mathbf{d}_j} \left[\sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \|\Delta_i[(\mathbf{x}_i - \mathbf{D}_{-j}\alpha_{i,-j}) - \mathbf{d}_j\alpha_{i,j}]\|_2^2 \right] \quad (\text{A.9})$$

$$= \frac{\partial}{\partial \mathbf{d}_j} \left[\sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \|(\Delta_i\alpha_{i,j})\mathbf{d}_j - \Delta_i(\mathbf{x}_i - \mathbf{D}_{-j}\alpha_{i,-j})\|_2^2 \right] \quad (\text{A.10})$$

$$= 2 \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i\alpha_{i,j} [(\Delta_i\alpha_{i,j})\mathbf{d}_j - \Delta_i(\mathbf{x}_i - \mathbf{D}_{-j}\alpha_{i,-j})] \quad (\text{A.11})$$

$$= 2 \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i\alpha_{i,j}^2 \mathbf{d}_j - 2 \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i\alpha_{i,j}(\mathbf{x}_i - \mathbf{D}_{-j}\alpha_{i,-j}), \quad (\text{A.12})$$

where we used the facts that

$$\mathbf{x}_{O_i} - \mathbf{D}_{O_i}\alpha_i = \Delta_i(\mathbf{x}_i - \mathbf{D}\alpha_i), \quad (\text{A.13})$$

$$\frac{\partial \|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2^2}{\partial \mathbf{y}} = 2\mathbf{A}^T(\mathbf{A}\mathbf{y} - \mathbf{b}), \quad (\text{A.14})$$

$$\Delta_i = \Delta_i^T = (\Delta_i)^2. \quad (\text{A.15})$$

After rearranging the terms in (A.12), we have that

$$\left(\sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \alpha_{i,j}^2 \right) \mathbf{u}_j = \quad (\text{A.16})$$

$$= \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \alpha_{i,j} (\mathbf{x}_i - \mathbf{D}_{-j} \boldsymbol{\alpha}_{i,-j}) \quad (\text{A.17})$$

$$= \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \mathbf{x}_i \alpha_{i,j} - \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \mathbf{D}_{-j} \boldsymbol{\alpha}_{i,-j} \alpha_{i,j} \quad (\text{A.18})$$

$$= \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \mathbf{x}_i \alpha_{i,j} - \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i (\mathbf{D}_{-j} \boldsymbol{\alpha}_{i,-j} + \mathbf{d}_j \alpha_{i,j} - \mathbf{d}_j \alpha_{i,j}) \alpha_{i,j} \quad (\text{A.19})$$

$$= \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \mathbf{x}_i \alpha_{i,j} - \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \mathbf{D} \boldsymbol{\alpha}_i \alpha_{i,j} + \left(\sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \alpha_{i,j}^2 \right) \mathbf{d}_j. \quad (\text{A.20})$$

We note that (A.18) is a system of linear equations, and its solution \mathbf{u}_j does not depend on \mathbf{d}_j . We have introduced the ‘ $\mathbf{d}_j \alpha_{i,j} - \mathbf{d}_j \alpha_{i,j}$ ’ term only for one purpose; it can help us with deriving the recursive updates for \mathbf{u}_j in a simple form. Define the following quantities

$$\mathbf{C}_{j,t} = \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \alpha_{i,j}^2 \in \mathbb{R}^{d_x \times d_x} \quad (j = 1, \dots, d_\alpha), \quad (\text{A.21})$$

$$\mathbf{B}_t = \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \mathbf{x}_i \boldsymbol{\alpha}_i^T = [\mathbf{b}_{1,t}, \dots, \mathbf{b}_{d_\alpha,t}] \in \mathbb{R}^{d_x \times d_\alpha}, \quad (\text{A.22})$$

$$\mathbf{e}_{j,t} = \sum_{i=1}^t \left(\frac{i}{t} \right)^\rho \Delta_i \mathbf{D} \boldsymbol{\alpha}_i \alpha_{i,j} \in \mathbb{R}^{d_x} \quad (j = 1, \dots, d_\alpha). \quad (\text{A.23})$$

Here (i) $\mathbf{C}_{j,t}$ s are diagonal matrices and (ii) the update rule of \mathbf{B}_t contains the quantity $\Delta_i \mathbf{x}_i$, which is \mathbf{x}_{O_i} extended by zeros at the non-observable ($\{1, \dots, d_x\} \setminus O_i$) coordinates. By using these notations and (A.20), we obtain that \mathbf{u}_j satisfies the following equation:

$$\mathbf{C}_{j,t} \mathbf{u}_j = \mathbf{b}_{j,t} - \mathbf{e}_{j,t} + \mathbf{C}_{j,t} \mathbf{d}_j. \quad (\text{A.24})$$

Now, according to Lemma 1, we can see that (i) when $\rho = 0$ and $\mathbf{C}_{j,0} = \mathbf{0}$, $\mathbf{B}_0 = \mathbf{0}$, or (ii) $\rho > 0$ and $\mathbf{C}_{j,0}$, \mathbf{B}_0 are arbitrary, then the $\mathbf{C}_{j,t}$ and \mathbf{B}_t quantities can be updated online with the following recursions:

$$\mathbf{C}_{j,t} = \gamma_t \mathbf{C}_{j,t-1} + \Delta_t \alpha_{t,j}^2, \quad (\text{A.25})$$

$$\mathbf{B}_t = \gamma_t \mathbf{B}_{t-1} + \Delta_t \mathbf{x}_t \alpha_t^T, \quad (\text{A.26})$$

where $\gamma_t = (1 - \frac{1}{t})^\rho$. We use the following online approximation for $\mathbf{e}_{j,t}$:

$$\mathbf{e}_{j,t} = \gamma_t \mathbf{e}_{j,t-1} + \Delta_t \mathbf{D} \alpha_t \alpha_{t,j}, \quad (\text{A.27})$$

with initialization $\mathbf{e}_{j,0} = \mathbf{0}$ ($\forall j$), and \mathbf{D} is the *actual* estimation for the dictionary. This choice seems to be efficient according to our numerical experiences.

Note. In the fully observable special case (i.e., when $\Delta_i = \mathbf{I}$, $\forall i$) the (A.21)-(A.23) equations have the following simpler form:

$$\mathbf{C}_{j,t} = \mathbf{I} \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \alpha_{i,j}^2, \quad (\text{A.28})$$

$$\mathbf{B}_t = \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \mathbf{x}_i \alpha_i^T, \quad (\text{A.29})$$

$$\mathbf{e}_{j,t} = \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \mathbf{D} \alpha_i \alpha_{i,j} = \mathbf{D} \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \alpha_i \alpha_{i,j}. \quad (\text{A.30})$$

Define the following term:

$$\mathbf{A}_t = \sum_{i=1}^t \left(\frac{i}{t}\right)^\rho \alpha_i \alpha_i^T \in \mathbb{R}^{d_\alpha \times d_\alpha}, \quad (\text{A.31})$$

and let $\mathbf{a}_{j,t}$ denote the j^{th} column of \mathbf{A}_t . Now, (A.30) can be rewritten as

$$\mathbf{e}_{j,t} = \mathbf{D} \mathbf{a}_{j,t}, \quad (\text{A.32})$$

and thus (A.24) has the following simpler form:

$$(\mathbf{A}_t)_{j,j} \mathbf{u}_j = \mathbf{b}_{j,t} - \mathbf{D} \mathbf{a}_{j,t} + (\mathbf{A}_t)_{j,j} \mathbf{d}_j. \quad (\text{A.33})$$

Here $(\cdot)_{j,j}$ stands for the $(j, j)^{\text{th}}$ entry of its argument. By applying again Lemma 1 for (A.31), we have that when (i) $\rho = 0$ and $\mathbf{A}_0 = \mathbf{0}$, or (ii) $\rho > 0$ and \mathbf{A}_0 is arbitrary, then \mathbf{A}_t can be updated online with the following recursion:

$$\mathbf{A}_t = \gamma_t \mathbf{A}_{t-1} + \alpha_t \alpha_t^T. \quad (\text{A.34})$$

We also note that in the fully observable case (A.26) reduces to

$$\mathbf{B}_t = \gamma_t \mathbf{B}_{t-1} + \mathbf{x}_t \alpha_t^T, \quad (\text{A.35})$$

and thus [77] is indeed a special case of our model:

- We calculate \mathbf{u}_j by (A.33).
- To optimize \hat{f}_t , it is enough to keep track of \mathbf{A}_t and \mathbf{B}_t instead of $\{\mathbf{C}_{j,t}\}_{j=1}^{d_\alpha}, \mathbf{B}_t, \{\mathbf{e}_{j,t}\}_{j=1}^{d_\alpha}$.
- The quantities \mathbf{A}_t and \mathbf{B}_t can be updated online by (A.34) and (A.35).

A.2 Correspondence of the (c, e)-SVM and (p, s)-Sparse Problems

In this section we give the proof of Proposition 1.

We will use the fact that the Moore-Penrose generalized inverse of a matrix $\mathbf{G} \in \mathbb{R}^{n \times m}$, $\mathbf{G}^- \in \mathbb{R}^{m \times n}$ uniquely exists and it has the properties:

$$\mathbf{G} \mathbf{G}^-, \mathbf{G}^- \mathbf{G} : \text{symmetric matrices} \quad (\text{A.36})$$

$$\mathbf{G} \mathbf{G}^- \mathbf{G} = \mathbf{G} \quad (\text{A.37})$$

$$\mathbf{G}^- \mathbf{G} \mathbf{G}^- = \mathbf{G}^-. \quad (\text{A.38})$$

We modify Eq. (2.50) using the assumption that $f(\mathbf{x}_i) = y_i$ ($i = 1, \dots, l$). Exploiting that for the norm $\|\cdot\|_{\mathcal{H}}^2 = \langle \cdot, \cdot \rangle_{\mathcal{H}}$ holds, and that scalar products are

bilinear we obtain

$$\begin{aligned}
F(\mathbf{a}) &= \frac{1}{2} \|f\|_{\mathcal{H}}^2 - \sum_{i=1}^l a_i \langle f(\cdot), k(\cdot, \mathbf{x}_i) \rangle_{\mathcal{H}} \\
&\quad + \frac{1}{2} \sum_{i,j=1}^l a_i a_j \langle k(\cdot, \mathbf{x}_i), k(\cdot, \mathbf{x}_j) \rangle_{\mathcal{H}} + \sum_{i=1}^l p_i |a_i|_{s_i}.
\end{aligned} \tag{A.39}$$

According to the reproducing property of the kernel, and our $f(\mathbf{x}_i) = y_i$ ($i = 1, \dots, l$) assumption, one can see that

$$\langle f(\cdot), k(\cdot, \mathbf{x}_i) \rangle_{\mathcal{H}} = f(\mathbf{x}_i) = y_i, \tag{A.40}$$

$$\langle k(\cdot, \mathbf{x}_i), k(\cdot, \mathbf{x}_j) \rangle_{\mathcal{H}} = k(\mathbf{x}_i, \mathbf{x}_j) = G_{ij}. \tag{A.41}$$

By dropping the first term of $F(\mathbf{a})$, which is independent of \mathbf{a} , we get that the minimization of $F(\mathbf{a})$ is equivalent to

$$\frac{1}{2} \mathbf{a}^T \mathbf{G} \mathbf{a} - \mathbf{y}^T \mathbf{a} + \sum_{i=1}^l p_i |a_i|_{s_i} \rightarrow \min_{\mathbf{a} \in \mathbb{R}^l}, \tag{A.42}$$

where $\mathbf{G} = [G_{ij}] = [k(\mathbf{x}_i, \mathbf{x}_j)]$ is the Gram matrix of the $\{\mathbf{x}_i\}$ samples. By rewriting the s_i -insensitive terms introducing slack variables, and introducing the notation $\mathbf{s} = [s_1; \dots; s_l]$, the optimization problem (A.42) is equivalent to

$$\begin{aligned}
&\min_{\mathbf{a}, \mathbf{s}^+, \mathbf{s}^-} \left[\frac{1}{2} \mathbf{a}^T \mathbf{G} \mathbf{a} - \mathbf{y}^T \mathbf{a} + \mathbf{p}^T (\mathbf{s}^+ + \mathbf{s}^-) \right], \\
&\text{subject to } \begin{cases} \mathbf{a} \leq \mathbf{s} + \mathbf{s}^+ \\ -\mathbf{a} \leq \mathbf{s} + \mathbf{s}^- \\ \mathbf{0} \leq \mathbf{s}^+, \mathbf{s}^- \end{cases}.
\end{aligned} \tag{A.43}$$

Now we take the dual of this problem using the Lagrangian approach

$$\begin{aligned}
&\max_{\mathbf{d}^+, \mathbf{d}^-, \mathbf{q}^+, \mathbf{q}^- \geq 0} L(\mathbf{d}^+, \mathbf{d}^-, \mathbf{q}^+, \mathbf{q}^-) = \\
&= \frac{1}{2} \mathbf{a}^T \mathbf{G} \mathbf{a} - \mathbf{y}^T \mathbf{a} + \mathbf{p}^T (\mathbf{s}^+ + \mathbf{s}^-) - (\mathbf{q}^+)^T \mathbf{s}^+ - (\mathbf{q}^-)^T \mathbf{s}^- \\
&\quad - (\mathbf{d}^+)^T (\mathbf{s} + \mathbf{s}^+ - \mathbf{a}) - (\mathbf{d}^-)^T (\mathbf{s} + \mathbf{s}^- + \mathbf{a}).
\end{aligned} \tag{A.44}$$

At the optimum, the derivatives of Lagrangian L taken by the primal variables disappear, that is

$$\mathbf{0} = \frac{\partial L}{\partial \mathbf{a}} = \mathbf{a}^T \mathbf{G} - \mathbf{y}^T + (\mathbf{d}^+ - \mathbf{d}^-)^T, \quad (\text{A.45})$$

$$\mathbf{0} = \frac{\partial L}{\partial \mathbf{s}^+} = \mathbf{p}^T - (\mathbf{d}^+)^T - (\mathbf{q}^+)^T, \quad (\text{A.46})$$

$$\mathbf{0} = \frac{\partial L}{\partial \mathbf{s}^-} = \mathbf{p}^T - (\mathbf{d}^-)^T - (\mathbf{q}^-)^T. \quad (\text{A.47})$$

Reordering and transposing (A.45), we have

$$\mathbf{a}^T \mathbf{G} = (\mathbf{y} - (\mathbf{d}^+ - \mathbf{d}^-))^T, \quad (\text{A.48})$$

$$\mathbf{G} \mathbf{a} = (\mathbf{y} - (\mathbf{d}^+ - \mathbf{d}^-)), \quad (\text{A.49})$$

where the symmetric property of Gram matrix \mathbf{G} was also exploited. Using (A.48), we can substitute expression

$$(\mathbf{y} - (\mathbf{d}^+ - \mathbf{d}^-))^T \mathbf{a} = \mathbf{a}^T \mathbf{G} \mathbf{a} \quad (\text{A.50})$$

to L . One can also replace matrix \mathbf{G} of the Lagrangian by $\mathbf{G} \mathbf{G}^- \mathbf{G}$ according to (A.37), and then insert the expressions for $\mathbf{a}^T \mathbf{G}$ and $\mathbf{G} \mathbf{a}$ using (A.48) and (A.49) to obtain

$$\mathbf{a}^T \mathbf{G} \mathbf{a} = \mathbf{a}^T (\mathbf{G} \mathbf{G}^- \mathbf{G}) \mathbf{a} = (\mathbf{a}^T \mathbf{G}) \mathbf{G}^- (\mathbf{G} \mathbf{a}) \quad (\text{A.51})$$

$$= (\mathbf{y} - (\mathbf{d}^+ - \mathbf{d}^-))^T \mathbf{G}^- (\mathbf{y} - (\mathbf{d}^+ - \mathbf{d}^-)). \quad (\text{A.52})$$

Using expressions (A.46) and (A.47) in the Lagrangian L , the variables $\mathbf{q}^+, \mathbf{q}^-$ disappear, but their non-negativity conditions, with (A.46) and (A.47) give rise to constraints $\mathbf{p} \geq \mathbf{d}^+$ and $\mathbf{p} \geq \mathbf{d}^-$ for variables \mathbf{d}^+ and \mathbf{d}^- . We can also change the minimization of Lagrangian L to maximization by changing the sign. Taken together, we have that our optimization task is that of

$$\min_{\mathbf{p} \geq \mathbf{d}^+, \mathbf{d}^- \geq \mathbf{0}} \left[\frac{1}{2} (\mathbf{y} - (\mathbf{d}^+ - \mathbf{d}^-))^T \mathbf{G}^- (\mathbf{y} - (\mathbf{d}^+ - \mathbf{d}^-)) + (\mathbf{d}^+ + \mathbf{d}^-)^T \mathbf{s} \right]. \quad (\text{A.53})$$

The terms of the quadratic expression can be expanded and reordered. Upon drop-

ping terms not containing variables \mathbf{d}^+ or \mathbf{d}^- , and making use of the symmetric property of \mathbf{G}^- inherited from \mathbf{G} , one obtains that the optimization problem is

$$\min_{\mathbf{p} \geq \mathbf{d}^+, \mathbf{d}^- \geq 0} \left[\frac{1}{2} (\mathbf{d}^+ - \mathbf{d}^-)^T \mathbf{G}^- (\mathbf{d}^+ - \mathbf{d}^-) - (\mathbf{d}^+ - \mathbf{d}^-)^T \mathbf{G}^- \mathbf{y} + (\mathbf{d}^+ + \mathbf{d}^-)^T \mathbf{s} \right]. \quad (\text{A.54})$$

Now, comparing the obtained result with (2.47), we can see that one can transform the dual of the (\mathbf{p}, \mathbf{s}) -sparse task to that of the (\mathbf{c}, \mathbf{e}) -SVM task according to the relation the $(\mathbf{d}^*, \mathbf{d}, \mathbf{G}, \mathbf{y}) \leftrightarrow (\mathbf{d}^+, \mathbf{d}^-, \mathbf{G}^-, \mathbf{G}^- \mathbf{y}) = (\mathbf{d}^+, \mathbf{d}^-, \mathbf{G}^- \mathbf{G} \mathbf{G}^-, \mathbf{G}^- \mathbf{y})$. At the last equality, the (A.38) property of the generalized inverse was used. This is what we wanted to prove. \square

A.3 Backpropagation for Multilayer Kerceptrons

In the sequel, we derive propagation rule for the multilayer kerceptron network. We carry out the derivation for steepest descent optimization. The cost function is has two terms: $c(t) = \varepsilon^2(t) + r(t)$. In Section A.3.1 we focus on the derivative of the $\varepsilon^2(t)$ approximation term. In Section A.3.2, we are dealing with the regularization part. The obtained results are embedded into steepest descent optimization in Section A.3.3.

A.3.1 Derivative of the Approximation Term

In this section we derive the derivative of the $\varepsilon^2(t)$ approximation term. First, we list basic relations, involved by the MLK structure. For the case of better readability, below, index t is dropped [precise form: $\mathbf{x}^l = \mathbf{x}^l(t)$, $\mathbf{y}^l = \mathbf{y}^l(t)$, $\mathbf{s}^l = \mathbf{s}^l(t)$, $\mathbf{w}_i^l = \mathbf{w}_i^l(t)$].

$$\mathbf{x}^l = \mathbf{y}^{l-1} \in \mathbb{R}^{N_l^l} \quad (l = 1, \dots, L+1), \quad (\text{A.55})$$

$$\mathbf{x}^{l+1} = \mathbf{g}^l(\mathbf{s}^l) \quad (l = 1, \dots, L), \quad (\text{A.56})$$

$$\mathbf{s}^l = \begin{bmatrix} \langle \mathbf{w}_1^l, \boldsymbol{\varphi}^l(\mathbf{x}^l) \rangle_{\mathcal{H}^l} \\ \vdots \\ \langle \mathbf{w}_i^l, \boldsymbol{\varphi}^l(\mathbf{x}^l) \rangle_{\mathcal{H}^l} \\ \vdots \end{bmatrix} \quad (l = 1, \dots, L; i = 1, \dots, N_S^l) \quad (\text{A.57})$$

$$= \begin{bmatrix} \langle \mathbf{w}_1^l, \boldsymbol{\varphi}^l(\mathbf{g}^{l-1}(\mathbf{s}^{l-1})) \rangle_{\mathcal{H}^l} \\ \vdots \\ \langle \mathbf{w}_i^l, \boldsymbol{\varphi}^l(\mathbf{g}^{l-1}(\mathbf{s}^{l-1})) \rangle_{\mathcal{H}^l} \\ \vdots \end{bmatrix} \quad (l = 2, \dots, L; i = 1, \dots, N_S^l), \quad (\text{A.58})$$

$$\mathbf{s}^{l+1} = \begin{bmatrix} \langle \mathbf{w}_1^{l+1}, \boldsymbol{\varphi}^{l+1}(\mathbf{g}^l(\mathbf{s}^l)) \rangle_{\mathcal{H}^{l+1}} \\ \vdots \\ \langle \mathbf{w}_i^{l+1}, \boldsymbol{\varphi}^{l+1}(\mathbf{g}^l(\mathbf{s}^l)) \rangle_{\mathcal{H}^{l+1}} \\ \vdots \end{bmatrix} \quad (l = 1, \dots, L-1; i = 1, \dots, N_S^{l+1}). \quad (\text{A.59})$$

Our goal is to compute the quantity $\frac{\partial[\varepsilon^2(t)]}{\partial[\mathbf{w}_i^l(t)]}$, which according to the chain rule and the definition of $\mathbf{s}^l(t)$ takes the form

$$\frac{\partial[\varepsilon^2(t)]}{\partial[\mathbf{w}_i^l(t)]} = \frac{\partial[\varepsilon^2(t)]}{\partial[s_i^l(t)]} \frac{\partial[s_i^l(t)]}{\partial[\mathbf{w}_i^l(t)]} = \delta_i^l(t) \boldsymbol{\varphi}^l(\mathbf{x}^l(t)) \quad (l = 1, \dots, L; i = 1, \dots, N_S^l), \quad (\text{A.60})$$

where $\delta_i^l(t)$ is the i^{th} coordinate of the backpropagated error of layer l defined as

$$\boldsymbol{\delta}^l(t) = \frac{\partial[\varepsilon^2(t)]}{\partial[\mathbf{s}^l(t)]} \quad (l = 1, \dots, L). \quad (\text{A.61})$$

Let us notice that the derivative (A.60) can be expressed by using quantity $\delta_i^l(t)$ and by the feature representation of the input $\mathbf{x}^l(t)$ arriving to the l^{th} layer, i.e., by $\boldsymbol{\varphi}^l(\mathbf{x}^l(t))$.

Making use of the chain rule again and the definition of $\boldsymbol{\delta}^{l+1}(t)$, the backpropa-

gated error satisfies the relation

$$\delta^l(t) = \frac{\partial[\varepsilon^2(t)]}{\partial[\mathbf{s}^l(t)]} = \frac{\partial[\varepsilon^2(t)]}{\partial[\mathbf{s}^{l+1}(t)]} \frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]} = \delta^{l+1}(t) \frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]} \quad (l = 1, \dots, L-1). \quad (\text{A.62})$$

One can compute this recursion for the backpropagated error, and thus the required derivative (A.60), provided that (i) $\delta^L(t)$ and (ii) $\frac{\partial[\varepsilon^2(t)]}{\partial[\mathbf{s}^L(t)]}$ are available. In the sequel, we focus on the computation of these two quantities.

The $\delta^L(t)$ quantity can be computed as follows:

$$\delta^L(t) = \frac{\partial[\varepsilon^2(t)]}{\partial[\mathbf{s}^L(t)]} = \frac{\partial \left[\|\mathbf{d}(t) - \mathbf{g}^L(\mathbf{s}^L(t))\|_2^2 \right]}{\partial[\mathbf{s}^L(t)]} \quad (\text{A.63})$$

$$= 2 [\mathbf{g}^L(\mathbf{s}^L(t)) - \mathbf{d}(t)]^T (\mathbf{g}^L)'(\mathbf{s}^L(t)) \quad (\text{A.64})$$

$$= 2 [\mathbf{y}(t) - \mathbf{d}(t)]^T (\mathbf{g}^L)'(\mathbf{s}^L(t)). \quad (\text{A.65})$$

Here we used the chain rule, the equation

$$\frac{\partial[\|\mathbf{d} - \mathbf{y}\|_2^2]}{\partial \mathbf{y}} = 2(\mathbf{y} - \mathbf{d})^T, \quad (\text{A.66})$$

and inserted the relation

$$\mathbf{y}(t) = \mathbf{g}^L(\mathbf{s}^L(t)), \quad (\text{A.67})$$

imposed by the MLK architecture.

To compute

$$\frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]} \quad (l = 1, \dots, L-1) \quad (\text{A.68})$$

(A.59) is made use of. It is sufficient to consider terms of the form

$$\frac{\partial[\langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s})) \rangle_{\mathcal{H}}]}{\partial[\mathbf{s}]} \quad (\text{A.69})$$

and then to ‘compile’ the full derivative from them. The value of (A.69) can computed by means of the following lemma.

Lemma 2. *Let $\mathbf{w} \in \mathcal{H} = \mathcal{H}(k)$ be a point in the RKHS \mathcal{H} . Let us assume the followings:*

1. *Explicit case: the $\mathbf{x} \mapsto \langle \mathbf{w}, \varphi(\mathbf{x}) \rangle_{\mathcal{H}}$ and the function \mathbf{g} are differentiable.*

2. *Implicit case:*

- *Let kernel k be differentiable w.r.t. both arguments and let k'_y denote the derivative of the kernel according to its second argument.*
- *We also assume that \mathbf{w} is within the image space of the feature space representation of a finite number of points \mathbf{z}_i . That is*

$$\mathbf{w} \in \text{Im}(\varphi(\mathbf{z}_1), \varphi(\mathbf{z}_2), \dots, \varphi(\mathbf{z}_N)) \subseteq \mathcal{H}. \quad (\text{A.70})$$

Let this expansion be $\mathbf{w} = \sum_{j=1}^N \alpha_j \varphi(\mathbf{z}_j)$, where $\alpha_j \in \mathbb{R}$.

Then we have two cases:

1. *Explicit case:*

$$\frac{\partial[\langle \mathbf{w}, \varphi(\mathbf{g}(\mathbf{s})) \rangle_{\mathcal{H}}]}{\partial[\mathbf{s}]} = \frac{\partial[\langle \mathbf{w}, \varphi(\mathbf{u}) \rangle_{\mathcal{H}}]}{\partial[\mathbf{u}]} \Big|_{\mathbf{u}=\mathbf{g}(\mathbf{s})} \mathbf{g}'(\mathbf{s}). \quad (\text{A.71})$$

2. *Implicit case:*

$$\frac{\partial[\langle \mathbf{w}, \varphi(\mathbf{g}(\mathbf{s})) \rangle_{\mathcal{H}}]}{d[\mathbf{s}]} = \sum_{j=1}^N \alpha_j k'_y(\mathbf{z}_j, \mathbf{g}(\mathbf{s})) \mathbf{g}'(\mathbf{s}). \quad (\text{A.72})$$

Proof.

1. *Explicit case: the statement follows from the chain rule.*

2. Implicit case:

$$\frac{\partial[\langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s})) \rangle_{\mathcal{H}}]}{\partial[\mathbf{s}]} = \frac{\partial \left[\left\langle \sum_j \alpha_j \boldsymbol{\varphi}(\mathbf{z}_j), \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s})) \right\rangle_{\mathcal{H}} \right]}{\partial[\mathbf{s}]} \quad (\text{A.73})$$

$$= \frac{\partial \left[\sum_j \alpha_j \langle \boldsymbol{\varphi}(\mathbf{z}_j), \boldsymbol{\varphi}(\mathbf{g}(\mathbf{s})) \rangle_{\mathcal{H}} \right]}{\partial[\mathbf{s}]} \quad (\text{A.74})$$

$$= \frac{\partial \left[\sum_j \alpha_j k(\mathbf{z}_j, \mathbf{g}(\mathbf{s})) \right]}{\partial[\mathbf{s}]} \quad (\text{A.75})$$

$$= \sum_j \alpha_j k'_y(\mathbf{z}_j, \mathbf{g}(\mathbf{s})) \mathbf{g}'(\mathbf{s}). \quad (\text{A.76})$$

In the first equation the expansion of \mathbf{w} and the linear property of the scalar product was exploited. Then, the relation (2.34) between the feature mapping and the kernel was applied. The last step follows from the chain rule.

□

Let us turn back to the computation of Eq. (A.68):

1. Explicit case: According to Lemma 2 we have

$$\frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]} = \begin{bmatrix} \vdots \\ \frac{\partial[\langle \mathbf{w}_i^{l+1}(t), \boldsymbol{\varphi}^{l+1}(\mathbf{u}) \rangle_{\mathcal{H}^{l+1}}]}{\partial[\mathbf{u}]} \Big|_{\mathbf{u}=\mathbf{g}^l(\mathbf{s}^l(t))} (\mathbf{g}^l)' (\mathbf{s}^l(t)) \\ \vdots \end{bmatrix} \quad (\text{A.77})$$

$$= \begin{bmatrix} \vdots \\ \frac{\partial[\langle \mathbf{w}_i^{l+1}(t), \boldsymbol{\varphi}^{l+1}(\mathbf{u}) \rangle_{\mathcal{H}^{l+1}}]}{\partial[\mathbf{u}]} \Big|_{\mathbf{u}=\mathbf{x}^{l+1}(t)} (\mathbf{g}^l)' (\mathbf{s}^l(t)) \\ \vdots \end{bmatrix} \quad (\text{A.78})$$

($l = 1, \dots, L-1; i = 1, \dots, N_S^{l+1}$).

In the second equation (i) we used identity (A.56) and (ii) pulled out the term $(\mathbf{g}^l)' (\mathbf{s}^l(t))$ according to the matrix multiplication rules.

2. Implicit case: For terms $\mathbf{w}_i^{l+1}(t)$ we have the expansion property expressed

by Eq. (2.58). This was our starting assumption. In subsection A.3.3, we shall see that this feature is ‘inherited’ from time (t) to time $(t + 1)$. Thus,

$$\mathbf{w}_i^{l+1}(t) = \sum_{j=1}^{N_i^{l+1}(t)} \alpha_{ij}^{l+1}(t) \boldsymbol{\varphi}^{l+1}(\mathbf{z}_{ij}^{l+1}(t)) \quad (l = 1, \dots, L-1; i = 1, \dots, N_S^{l+1}) \quad (\text{A.79})$$

and the derivative (A.68), we need, takes the form

$$\frac{\partial[\mathbf{s}^{l+1}(t)]}{\partial[\mathbf{s}^l(t)]} = \begin{bmatrix} \vdots \\ \sum_{j=1}^{N_i^{l+1}(t)} \alpha_{ij}^{l+1}(t) [k^{l+1}]'_y(\mathbf{z}_{ij}^{l+1}(t), \mathbf{g}^l(\mathbf{s}^l(t))) (\mathbf{g}^l)'(\mathbf{s}^l(t)) \\ \vdots \end{bmatrix} \quad (\text{A.80})$$

$$= \begin{bmatrix} \vdots \\ \sum_{j=1}^{N_i^{l+1}(t)} \alpha_{ij}^{l+1}(t) [k^{l+1}]'_y(\mathbf{z}_{ij}^{l+1}(t), \mathbf{x}^{l+1}(t)) (\mathbf{g}^l)'(\mathbf{s}^l(t)) \\ \vdots \end{bmatrix} \quad (\text{A.81})$$

$$(l = 1, \dots, L-1; i = 1, \dots, N_S^{l+1}).$$

Here, the second equation is based on identity (A.56). Matrix term $(\mathbf{g}^l)'(\mathbf{s}^l(t))$ was pulled out according to the matrix multiplication rules.

A.3.2 Derivative of the Regularization Term

The derivative of the regularization term $r(t)$ is simple:

$$\frac{\partial[r(t)]}{\partial[\mathbf{w}_i^l(t)]} = \frac{\partial \left[\sum_{l=1}^L \sum_{i=1}^{N_S^l} \lambda_i^l \|\mathbf{w}_i^l(t)\|_{\mathcal{H}^l}^2 \right]}{\partial[\mathbf{w}_i^l(t)]} = 2\lambda_i^l \mathbf{w}_i^l(t) \quad (l = 1, \dots, L; i = 1, \dots, N_S^l). \quad (\text{A.82})$$

Note that the respective terms of the derivative are scaled version of the actual weights, $\mathbf{w}_i^l(t)$. This form makes possible implicit tuning in the dual space.

A.3.3 Derivative of the Cost

Using identity

$$\frac{\partial[c(t)]}{\partial[\mathbf{w}_i^l(t)]} = \frac{\partial[\varepsilon^2(t)]}{\partial[\mathbf{w}_i^l(t)]} + \frac{\partial[r(t)]}{\partial[\mathbf{w}_i^l(t)]} \quad (l = 1, \dots, L; i = 1, \dots, N_S^l) \quad (\text{A.83})$$

as well as our results on the approximation and the regularization terms [i.e., Eqs. (A.60), and (A.82)], for the

$$\mathbf{w}_i^l(t+1) = \mathbf{w}_i^l(t) - \mu_i^l(t) \frac{\partial[c(t)]}{\partial[\mathbf{w}_i^l(t)]} \quad (l = 1, \dots, L; i = 1, \dots, N_S^l). \quad (\text{A.84})$$

steepest descent form we have

$$\mathbf{w}_i^l(t+1) = \mathbf{w}_i^l(t) - \mu_i^l(t) (\delta_i^l(t) \boldsymbol{\varphi}^l(\mathbf{x}^l(t)) + 2\lambda_i^l \mathbf{w}_i^l(t)) \quad (\text{A.85})$$

$$= (1 - 2\mu_i^l(t)\lambda_i^l) \mathbf{w}_i^l(t) - \mu_i^l(t) \delta_i^l(t) \boldsymbol{\varphi}^l(\mathbf{x}^l(t)) \quad (\text{A.86})$$

$(l = 1, \dots, L; i = 1, \dots, N_S^l).$

The same in dual form is as follows

$$\boldsymbol{\alpha}_i^l(t+1) = [(1 - 2\mu_i^l(t)\lambda_i^l) \boldsymbol{\alpha}_i^l(t); -\mu_i^l(t) \delta_i^l(t)] \quad (l = 1, \dots, L; i = 1, \dots, N_S^l). \quad (\text{A.87})$$

In turn, according to (A.86) the expansion property of the weight vectors of the network [i.e., Eq. (2.58)] is inherited from time (t) to time $(t+1)$. In particular, the expansion is valid for parameter set \mathbf{w}_i^l received at the end of the computation. To sum up, the backpropagation procedure holds for MLK. The derived explicit and implicit procedures are summarized in Table 2.5 and Table 2.6, respectively.

Appendix B

Abbreviations

Abbreviations used in the paper are listed in Table B.1.

Table B.1: Acronyms.

Abbreviation	Meaning
ANN	approximate nearest neighbor
AR	autoregressive
ARMA	autoregressive moving average
ARMAX	ARMA with exogenous input
ARX	AR with exogenous input
BCD	block coordinate descent
BCDA	approximate block coordinate descent
BSD	blind source deconvolution
BSSD	blind subspace deconvolution
CE	cross-entropy
CF	collaborative filtering
ECG	electro-cardiography
EEG	electro-encephalography
EM	expectation maximization
fAR	functional AR
fMRI	functional magnetic resonance imaging
ICA/ISA/IPA	independent component/subspace/process analysis

i.i.d.	independent identically distributed
JFD	joint f-decorrelation
KCCA	kernel canonical correlation analysis
Lasso	least absolute shrinkage and selection operator
LDS	linear dynamical system
LPA	linear prediction approximation
MA	moving average
mAR	AR with missing values
MEG	magneto-encephalography
ML	maximum likelihood
MLK	multilayer kerceptron
MLP	multilayer perceptron
MSE	mean square error
NIW	normal-inverted Wishart
NN	nearest neighbor
NMF	non-negative matrix factorization
OSDL	online group-structured dictionary learning
PCA	principal component analysis
PNL	post nonlinear
PSNR	peak signal-to-noise ratio
QP	quadratic programming
RADICAL	robust, accurate, direct ICA algorithm
RBF	radial basis function
RIP	restrictive isometry property
RMSE	root mean square error
RKHS	reproducing kernel Hilbert space
RP	random projection
RS	recommender system
SDL	structured dictionary learning
SVM	support vector machine
TSP	traveling salesman problem

Bibliography

- [1] Joel A. Tropp and Stephen J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE special issue on Applications of sparse representation and compressive sensing*, 98(6):948–958, 2010.
- [2] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. Technical report, 2010. <http://arxiv.org/abs/1012.0621>.
- [3] Balas K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.
- [4] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- [5] Junzhou Huang and Tong Zhang. The benefit of group sparsity. *Annals of Statistics*, 38(4):1978–2004, 2010.
- [6] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.
- [7] Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56:1982–2001, 2010.

- [8] Shaoting Zhang, Junzhou Huang, Yuchi Huang, Yang Yu, Hongsheng Li, and Dimitris N. Metaxas. Automatic image annotation using group sparsity. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3312–3319, 2010.
- [9] Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics*, 37(6A):3468–3497, 2009.
- [10] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group Lasso with overlap and graph Lasso. In *International Conference on Machine Learning (ICML)*, pages 433–440, 2009.
- [11] Seyoung Kim and Eric P. Xing. Tree-guided group Lasso for multi-task regression with structured sparsity. In *International Conference on Machine Learning (ICML)*, pages 543–550, 2010.
- [12] Lorenzo Rosasco, Sofia Mosci, Silvia Villa, and Alessandro Verri. A primal-dual algorithm for group sparse regularization with overlapping groups. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2604–2612, 2010.
- [13] Charles A. Mitchelli, Jean M. Morales, and Massimiliano Pontil. A family of penalty functions for structured sparsity. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1612–1623, 2010.
- [14] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. *Optimization for Machine Learning*, chapter Convex optimization with sparsity-inducing norms. MIT Press, 2011.
- [15] Franck Rapaport, Emmanuel Barillot, and Jean-Philippe Vert. Classification of arrayCGH data using fused SVM. *Bioinformatics*, 24(13):i375–i382, 2008.
- [16] Xi Chen, Qihang Lin, Seyoung Kim, Jaime G. Carbonell, and Eric P. Xing. An efficient proximal gradient method for general structured sparse learning. Technical report, 2010. <http://arxiv.org/abs/1005.4717>.

- [17] Jun Liu and Jieping Ye. Fast overlapping group lasso. Technical report, 2010. <http://arxiv.org/abs/1009.0306>.
- [18] Julien Chiquet, Yves Grandvalet, and Camille Charbonnier. Sparsity with sign-coherent groups of variables via the cooperative-Lasso. Technical report, 2011. <http://arxiv.org/abs/1103.2697>.
- [19] Haiqin Yang, Zenglin Xu, Irwin King, and Michael R. Lyu. Online learning for group Lasso. In *International Conference on Machine Learning (ICML)*, pages 1191–1198, 2010.
- [20] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.
- [21] Guillaume Obozinski, Ben Taskar, and Michael I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20:231–252, 2010.
- [22] Sofia Mosci, Lorenzo Rosasco, Matteo Santoro, Alessandro Verri, and Silvia Villa. Solving structured sparsity regularization with proximal methods. In *European Conference on Machine Learning (ECML)*, pages 418–433, 2010.
- [23] Pablo Sprechmann, Ignacio Ramírez, Guillermo Sapiro, and Yonina Eldar. C-HiLasso: A collaborative hierarchical sparse modeling framework. Technical report, 2010. <http://arxiv.org/abs/arXiv:1006.1346>.
- [24] Alain Rakotomamonjy. Review: Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms. *Signal Processing*, 91(7):1505–1526.
- [25] Hongtao Lu, Xianzhong Long, and Jingyuan Lv. A fast algorithm for recovery of jointly sparse vectors based on the alternating direction methods. *JMLR:W&CP*, 15, 2011.
- [26] Dongmin Kim, Suvrit Sra, and Inderjit S. Dhillon. A scalable trust-region algorithm with application to mixed-norm regression. In *International Conference on Machine Learning (ICML)*, pages 519–526, 2010.

- [27] Yonina C. Eldar and Moshe Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Transactions on Signal Processing*, 55(11):5302 – 5316, 2009.
- [28] Howard D. Bondell and Brian J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64(1):115–123, 2008.
- [29] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009.
- [30] André Martins, Noah Smith, Eric Xing, Pedro Aguiar, and Mario Figueiredo. Online learning of structured predictors with multiple kernels. *JMLR:W&CP*, 15, 2011.
- [31] Samy Bengio, Fernando Pereira, Yoram Singer, and Dennis Strelow. Group sparse coding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 82–89, 2009.
- [32] Jun Liu and Jieping Ye. Moreau-Yosida regularization for grouped tree structure learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1459–1467, 2010.
- [33] Ehsan Elhamifar and René Vidal. Robust classification using structured sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1873 – 1879, 2011.
- [34] Andreas Argyriou, Charles A. Micchelli, Massimiliano Pontil, Lixin Shen, and Yuesheng Xu. Efficient first order methods for linear composite regularizers. Technical report, 2011. <http://arxiv.org/abs/1104.1436>.
- [35] Subhojit Som, Lee C. Potter, and Philip Schniter. Compressive imaging using approximate message passing and a Markov-tree prior. In *Asilomar Conference on Signals, Systems and Computers*, 2010.
- [36] Philip Schniter. Turbo reconstruction of structured sparse signals. In *Conference on Information Sciences and Systems*, 2010.

- [37] Angshul Majumdar and Rabab K. Ward. Compressive color imaging with group-sparsity on analysis prior. In *International Conference on Image Processing (ICIP)*, pages 1337–1340, 2010.
- [38] Volkan Cevher and Sina Jafarpour. Fast hard thresholding with Nesterov’s gradient method. In *Advances in Neural Information Processing Systems (NIPS): Workshop on Practical Applications of Sparse Modeling*, 2010.
- [39] Mark Schmidt and Kevin Murphy. Convex structure learning in log-linear models: Beyond pairwise potentials. *JMLR:W&CP*, 9:709–716, 2010.
- [40] Ali Jalali, Pradeep Ravikumar, Vishvas Vasuki, and Sujay Sanghavi. On learning discrete graphical models using group-sparse regularization. *JMLR:W&CP*, 2011.
- [41] Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Factorized latent spaces with structured sparsity. In *Advances in Neural Information Processing Systems (NIPS)*, pages 982–990, 2010.
- [42] Rodolphe Jenatton, Alexandre Gramfort, Vincent Michel, Guillaume Obozinski, Evelyn Eger, Francis Bach, and Bertrand Thirion. Multi-scale mining of fMRI data with hierarchical structured sparsity. Technical report, 2011. http://hal.inria.fr/docs/00/58/97/85/PDF/sparse_hierarchical_fmri_mining_HAL.pdf.
- [43] Yiyuan She. Sparse regression with exact clustering. *Electronic Journal of Statistics*, 4:1055–1096, 2010.
- [44] Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. ℓ_p -norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, 2011.
- [45] Marie Szafranski, Yves Grandvalet, and Alain Rakotomamonjy. Composite kernel learning. *Machine Learning*, 79:73–103, 2010.
- [46] Jonathan Aflalo, Aharon Ben-Tal, Chiranjib Bhattacharyya, Jagarlapudi Saketha Nath, and Sankaran Raman. Variable sparsity kernel learning. *Journal of Machine Learning Research*, 12:565–592, 2011.

- [47] Alain Rakotomamonjy, Rémi Flamary, Gilles Gasso, and Stéphane Canu. $\ell_p - \ell_q$ penalty for sparse linear and multiple kernel multi-task learning. Technical report, 2010. <http://hal.archives-ouvertes.fr/docs/00/50/96/08/PDF/mtlfeatsel.pdf>.
- [48] Francis Bach, Gert Lanckriet, and Michael Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *International Conference on Machine Learning*, pages 41–48, 2004.
- [49] Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity inducing norms. Technical report, 2010. <http://arxiv.org/abs/0904.3523>.
- [50] Sina Hamidi Ghalehjeh, Massoud Babaie-Zadeh, and Christian Jutten. Fast block-sparse decomposition based on SL0. In *International Conference on Latent Variable Analysis and Signal Separation (ICA/LVA)*, pages 426–433, 2010.
- [51] Arvind Ganesh, Zihan Zhou, and Yi Ma. Separation of a subspace-sparse signal: Algorithms and conditions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3141–3144, 2009.
- [52] Francis Bach. Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 118–126, 2010.
- [53] Francis Bach. Shaping level sets with submodular functions. Technical report, 2010. http://hal.archives-ouvertes.fr/docs/00/54/29/49/PDF/symmsub_hal.pdf.
- [54] Aurélie C. Lozano, Grzegorz Świrszcz, and Naoki Abe. Grouped orthogonal matching pursuit for variable selection and prediction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1150–1158, 2009.
- [55] Yue M. Lu and Minh N. Do. A theory for sampling signals from a union of subspaces. *IEEE Transactions on Signal Processing*, 56(6):2334 – 2345, 2008.

- [56] Thomas Blumensath and Mike E. Davies. Sampling theorems for signals from the union of finite-dimensional linear subspaces. *IEEE Transactions on Information Theory*, 55(4):1872–1882, 2009.
- [57] Volkan Cevher. An ALPS view of sparse recovery. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.
- [58] Yonina C. Eldar, Patrick Kuppinger, and Helmut Bölcskei. Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Transactions on Signal Processing*, 58(6):3042 – 3054, 2010.
- [59] Ryan Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *Annals of Statistics*, 2011. (to appear).
- [60] Ryota Tomioka, Taiji Suzuki, and Masashi Sugiyama. Super-linear convergence of dual augmented-Lagrangian algorithm for sparsity regularized estimation. *Journal of Machine Learning Research*, 12:1349–1398, 2011.
- [61] Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.
- [62] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for sparse hierarchical dictionary learning. In *International Conference on Machine Learning (ICML)*, pages 487–494, 2010.
- [63] Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. *JMLR:W&CP*, 9:366–373.
- [64] Julien Mairal, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1558–1566, 2010.
- [65] Kevin Rosenblum, Lihi Zelnik-Manor, and Yonina Eldar. Dictionary optimization for block-sparse representations. In *AAAI Fall 2010 Symposium on Manifold Learning*, 2010.

- [66] Koray Kavukcuoglu, Marc'Aurelio Ranzato, Rob Fergus, and Yann LeCun. Learning invariant features through topographic filter maps. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1605–1612, 2009.
- [67] Jorge Silva, Minhua Chen, Yonina C. Eldar, Guillermo Sapiro, and Lawrence Carin. Blind compressed sensing over a structured union of subspaces. Technical report, 2011. <http://arxiv.org/abs/1103.2469>.
- [68] Patrick L. Combettes and Jean-Christophe Pesquet. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, volume 49, chapter Proximal splitting methods in signal processing. Springer, 2010.
- [69] Rick Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE, Signal Processing Letters*, 14(10):707 – 710, 2007.
- [70] Rick Chartrand and Valentina Staneva. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems*, 24:1–14, 2008.
- [71] Rayan Saab and Özgür Yılmaz. Sparse recovery by non-convex optimization – instance optimality. *Applied and Computational Harmonic Analysis*, 29:30–48, 2010.
- [72] Simon Foucart and Ming-Jun Lai. Sparsest solutions of underdetermined linear systems via ℓ_q -minimization for $0 < q \leq 1$. *Applied and Computational Harmonic Analysis*, 26:395–407, 2010.
- [73] Gilles Gasso, Alain Rakotomamonjy, and Stéphane Canu. Recovering sparse signals with a certain family of nonconvex penalties and DC programming. *IEEE Transactions on Signal Processing*, 57(12):4686 – 4698, 2009.
- [74] Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010.
- [75] Joshua Trzasko and Armando Manduca. Relaxed conditions for sparse signal recovery with general concave priors. *IEEE Transactions on Signal Processing*, 57(11):4347–4354, 2009.

- [76] Ignacio Ramirez and Guillermo Sapiro. Universal regularizers for robust sparse coding and modeling. Technical report, 2010. <http://arxiv.org/abs/1003.2941>.
- [77] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:10–60, 2010.
- [78] Karl Engan and Kjersti Skretting. Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing*, 58(4):2121 – 2130, 2010.
- [79] Léon Bottou and Yann LeCun. On-line learning for very large data sets. *Applied Stochastic Models in Business and Industry - Statistical Learning*, 21(2):137–151, 2005.
- [80] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul Kantor. *Recommender Systems Handbook*. Springer, 2011.
- [81] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Proceedings of the 48th Annual Allerton Conference*, 2010.
- [82] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.
- [83] Federico Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998.
- [84] Su-Yun Huang and Yuh-Jye Lee. Equivalence relations between support vector machines, sparse approximation, Bayesian regularization and Gauss-Markov prediction. Technical report, 2003.
- [85] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- [86] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

- [87] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008.
- [88] Simon Haykin. *Neural Networks*. Prentice Hall, New Jersey, USA, 1999.
- [89] Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.
- [90] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.
- [91] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
- [92] Jean-François Cardoso. Multidimensional independent component analysis. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1941–1944, 1998.
- [93] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. Fetal electrocardiogram extraction by source subspace separation. In *IEEE SP/Athos Workshop on Higher-Order Statistics*, pages 134–138, 1995.
- [94] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 556–562, 2000.
- [95] Botond Szatmáry, Barnabás Póczos, Julian Eggert, Edgar Körner, and András Lőrincz. Non-negative matrix factorization extended by sparse code shrinkage and by weight sparsification. In *European Conference on Artificial Intelligence (ECAI)*, pages 503–507, 2002.
- [96] Patrick Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

- [97] Christian Jutten and Jeanny Héroult. Blind separation of sources: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.
- [98] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.
- [99] Marian Stewart Bartlett, Javier R. Movellan, and Terrence J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6):1450–1464, 2002.
- [100] Pong C. Yuen and J. H. Lai. Face representation using independent component analysis. *Pattern Recognition*, 35(6):1247–1257, 2002.
- [101] Huzefa Neemuchwala, Alfred Hero, and Paul Carson. Image matching using alpha-entropy measures and entropic graphs. *Signal Processing*, 85(2):277–296, 2005.
- [102] Anthony J. Bell and Terrence J. Sejnowski. The ‘independent components’ of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997.
- [103] Robert Jenssen and Torbjorn Eltoft. Independent component analysis for texture segmentation. *Pattern Recognition*, 36(13):2301–2315, 2003.
- [104] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.
- [105] Andrzej Cichocki and Shun-ichi Amari. *Adaptive blind signal and image processing*. John Wiley & Sons, 2002.
- [106] Seungjin Choi, Andrzej Cichocki, Hyung-Min Park, and Soo-Yound Lee. Blind source separation and independent component analysis. *Neural Information Processing - Letters and Reviews*, 6:1–57, 2005.
- [107] Aapo Hyvärinen and Patrik O. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12:1705–1720, 2000.

- [108] Heyjin Kim, Seungjin Choi, and Sung-Yang Bang. Membership scoring via independent feature subspace analysis for grouping co-expressed genes. In *International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 1690–1695, 2003.
- [109] Alok Sharma and Kuldip K. Paliwal. Subspace independent component analysis using vector kurtosis. *Pattern Recognition*, 39:2227–2232, 2006.
- [110] Fabian J. Theis. Blind signal separation into groups of dependent signals using joint block diagonalization. In *International Society for Computer Aided Surgery (ISCAS)*, volume 6, pages 5878–5881, 2005.
- [111] Shotaro Akaho, Yasuhiko Kiuchi, and Shinji Umeyama. MICA: Multimodal independent component analysis. In *International Joint Conference on Neural Networks (IJCNN)*, volume 2, pages 927–932, 1999.
- [112] Florian Kohl, Gerd Wübbeler, Dorothea Kolossa, Clemens Elster, Markus Bär, and Reinhold Orglmeister. Non-independent BSS: A model for evoked MEG signals with controllable dependencies. In *Independent Component Analysis and Signal Separation (ICA)*, volume 5441, pages 443–450, 2009.
- [113] Sai Ma, Xi-Lin Li, Nicolle Correa, Tülay Adalı, and Vince Calhoun. Independent subspace analysis with prior information for fMRI data. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1922–1925, 2010.
- [114] Fabian Sinz, Eero Simoncelli, and Matthias Bethge. Hierarchical modeling of local image features through L_p -nested symmetric distributions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1696–1704, 2009.
- [115] Jong Kyoung Kim and Seungjin Choi. Tree-dependent components of gene expression data for clustering. In *International Conference on Artificial Neural Networks (ICANN)*, volume 4132, pages 837–846, 2006.
- [116] Stan Z. Li, XiaoGuang Lv, and HongJiang Zhang. View-subspace analysis of multi-view face patterns. In *IEEE ICCV Workshop on Recognition, Analysis,*

- and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS)*, pages 125–132, 2001.
- [117] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. Fetal electrocardiogram extraction by blind source subspace separation. *IEEE Transactions on biomedical engineering*, 47(5):567–572, 2000.
 - [118] Harald Stögbauer, Alexander Kraskov, Sergey A. Astakhov, and Peter Grassberger. Least dependent component analysis based on mutual information. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 70(066123), 2004.
 - [119] M. P. S. Chawla. Computational and mathematical methods in medicine. *Detection of indeterminacies in corrected ECG signals using parameterized multidimensional independent component analysis*, 10(2):85–115, 2009.
 - [120] Zhimin Fan, Jie Zhou, and Ying Wu. Motion segmentation based on independent subspace analysis. In *Asian Conference on Computer Vision (ACCV)*, 2004.
 - [121] Michael A. Casey and Alex Westner. Separation of mixed audio sources by independent subspace analysis. In *International Computer Music Conference (ICMC)*, pages 154–161, 2000.
 - [122] Carlos Silva Santos, João Eduardo Kögler Jr., and Emílio Del Moral Hernandez. Using independent subspace analysis for selecting filters used in texture processing. In *International Conference on Image Processing (ICIP)*, volume 1, pages 465–468, 2005.
 - [123] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Undercomplete blind subspace deconvolution. *Journal of Machine Learning Research*, 8:1063–1095, 2007.
 - [124] Godfrey H. Hardy and Srinivasa I. Ramanujan. Asymptotic formulae in combinatory analysis. *Proceedings of the London Mathematical Society*, 17(1):75–115, 1918.

- [125] James V. Uspensky. Asymptotic formulae for numerical functions which occur in the theory of partitions. *Bulletin of the Russian Academy of Sciences*, 14(6):199–218, 1920.
- [126] Barnabás Póczos, Zoltán Szabó, Melinda Kiszlinger, and András Lőrincz. Independent process analysis without a priori dimensional information. In *Independent Component Analysis and Blind Signal Separation (ICA 2007)*, pages 252–259, 2007.
- [127] Anisse Taleb and Christian Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, 10(47):2807–2820, 1999.
- [128] Christian Jutten and Juha Karhunen. Advances in blind source separation (BSS) and independent component analysis (ICA) for nonlinear systems. *International Journal of Neural Systems*, 14(5):267–292, 2004.
- [129] Jean-François Cardoso and Antoine Souloumiac. Blind beamforming for non-Gaussian signals. *IEE-Proceedings-F*, 140(6):362–370, 1993.
- [130] Jörn Anemüller, Terrence J. Sejnowski, and Scott Makeig. Complex independent component analysis of frequency-domain electroencephalographic data. *Neural Networks*, 16:1311–1323, 2003.
- [131] Vince D. Calhoun and Tülay Adalı. Complex infomax: Convergence and approximation of infomax with complex nonlinearities. *VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 44(1-2):173–190, 2006.
- [132] Jörn Anemüller, Jeng-Ren Duann, Terrence J. Sejnowski, and Scott Makeig. Spatio-temporal dynamics in fMRI recordings revealed with complex independent component analysis. *Neurocomputing*, 69(13-15):1502–1512, 2006.
- [133] Kwokleung Chan, Te-Won Lee, and Terrence J. Sejnowski. Variational Bayesian learning of ICA with missing data. *Neural Computation*, 15(8):1991 – 2011, 2003.

- [134] A. Taylan Cemgil, Cédric Févotte, and Simon J. Godsill. Variational and stochastic inference for Bayesian source separation. *Digital Signal Processing*, 17:891–913, 2007.
- [135] Michael S. Pedersen, Jan Larsen, Ulrik Kjems, and Lucas C. Parra. A survey of convolutive blind source separation methods. In *Springer Handbook of Speech Processing*. Springer Press, 2007.
- [136] Jörn Anemüller. Second-order separation of multidimensional sources with constrained mixing system. In *Independent Component Analysis and Blind Signal Separation (ICA)*, pages 16–23, 2006.
- [137] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Online group-structured dictionary learning. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2865–2872, 2011.
- [138] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Online dictionary learning with group structure inducing norms. In *International Conference on Machine Learning (ICML) – Structured Sparsity: Learning and Inference Workshop*, 2011. (accepted).
- [139] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.
- [140] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005.
- [141] Robert Tibshirani and Michael Saunders. Sparsity and smoothness via the fused Lasso. *Journal of the Royal Statistical Society, Series B*, 67(1):91–108, 2005.
- [142] Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dmitry Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.
- [143] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, 2004.

- [144] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1 -ball for learning in high dimensions. In *International Conference on Machine Learning (ICML)*, pages 272–279, 2008.
- [145] Ewout van den Berg, Mark Schmidt, Michael P. Friedlander, and Kevin Murphy. Group sparsity via linear-time projection. Technical report, 2008. http://www.optimization-online.org/DB_FILE/2008/07/2056.pdf.
- [146] Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.
- [147] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific Belmont, 1999.
- [148] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *Computational Learning Theory (COLT)*, pages 416–426, 2001.
- [149] Zoltán Szabó and András Lőrincz. ϵ -sparse representations: Generalized sparse approximation and the equivalent family of SVM tasks. *Acta Cybernetica*, 17(3):605–614, 2006.
- [150] Zoltán Szabó and András Lőrincz. Multilayer kerceptron [Többrétegű kerceptron]. *Journal of Applied Mathematics [Alkalmazott Matematikai Lapok]*, 24:209–222, 2007.
- [151] Barnabás Póczos and András Lőrincz. Identification of recurrent neural networks by Bayesian interrogation techniques. *Journal of Machine Learning Research*, 10:515–554, 2009.
- [152] Zoltán Szabó and András Lőrincz. Towards independent subspace analysis in controlled dynamical systems. In *ICA Research Network Workshop (ICARN)*, pages 9–12, 2008.
- [153] Fabian J. Theis. Towards a general independent subspace analysis. In *Advances in Neural Information Processing Systems (NIPS)*, volume 19, pages 1361–1368, 2007.

- [154] V.V. Petrov. Central limit theorem for m -dependent variables. In *Proceedings of the All-Union Conference on Probability Theory and Mathematical Statistics*, pages 38–44, 1958.
- [155] Zoltán Szabó and András Lőrincz. Controlled complete ARMA independent process analysis. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3038–3045, 2009.
- [156] Zoltán Szabó and András Lőrincz. Post nonlinear hidden infomax identification [Poszt nemlineáris rejtett infomax identifikáció]. In *Joint Conference of Hungarian PhD students [Tavaszi Szél Konferencia]*, pages 52–58, 2008.
- [157] Zoltán Szabó. Independent subspace analysis in case of missing observations [Hiányosan megfigyelt független altér analízis]. In *Symposium of Intelligent Systems [Intelligens Rendszerek–Fiatal Kutatók Szimpóziuma]*, 2009.
- [158] Zoltán Szabó. Autoregressive independent process analysis with missing observations. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 159–164, 2010.
- [159] Barnabás Póczos, Bálint Takács, and András Lőrincz. Independent subspace analysis on innovations. In *European Conference on Machine Learning (ECML)*, volume 3720, pages 698–706, 2005.
- [160] Jan Eriksson. Complex random vectors and ICA models: Identifiability, uniqueness and separability. *IEEE Transactions on Information Theory*, 52(3), 2006.
- [161] P. Krishnaiah and Jugan Lin. Complex elliptically symmetric distributions. *Communications in Statistics*, 15(12):3693–3718, 1986.
- [162] Zoltán Szabó and András Lőrincz. Real and complex independent subspace analysis by generalized variance. In *ICA Research Network International Workshop (ICARN)*, pages 85–88, 2006.
- [163] Zoltán Szabó and András Lőrincz. Complex independent process analysis. *Acta Cybernetica*, 19:177–190, 2009.

- [164] Zoltán Szabó and Barnabás Póczos. Nonparametric independent process analysis. In *European Signal Processing Conference (EUSIPCO)*, 2011. (accepted).
- [165] Denis Bosq. *Nonparametric Statistics for Stochastic Processes: Estimation and Prediction*. Lecture Notes in Statistics. Springer, 1998.
- [166] Nadine Hilgert and Bruno Portier. Strong uniform consistency and asymptotic normality of a kernel based error density estimator in functional autoregressive models. Technical report, 2009. <http://arxiv.org/abs/0905.2327v1>.
- [167] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Undercomplete blind subspace deconvolution via linear prediction. In *European Conference on Machine Learning (ECML)*, pages 740–747, 2007.
- [168] Ravikiran Rajagopal and Lee C. Potter. Multivariate MIMO FIR inverses. *IEEE Transactions on Image Processing*, 12:458 – 465, 2003.
- [169] Zoltán Szabó. Complete blind subspace deconvolution. In *Independent Component Analysis and Signal Separation (ICA)*, pages 138–145, 2009.
- [170] Wayne A. Fuller. *Introduction to Statistical Time Series*. Wiley-Interscience, 1995.
- [171] John Galbraith, Aman Ullah, and Victoria Zinde-Walsh. Estimation of the vector moving average model by vector autoregression. *Econometric Reviews*, 21(2):205–219, 2002.
- [172] Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. Estimation of Rényi entropy and mutual information based on generalized nearest-neighbor graphs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1849–1857, 2010.
- [173] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45(6):891 – 923, 1998.

- [174] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.
- [175] Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projections. *Machine Learning*, 63:161–182, 2006.
- [176] Jiří Matoušek. On variants of the Johnson-Lindenstrauss lemma. *Random Structures and Algorithms*, 33(2):142–156, 2008.
- [177] Santosh S. Vempala. *The Random Projection Method (DIMACS Series in Discrete Math)*, volume 65. 2005. <http://dimacs.rutgers.edu/Volumes/Vol65.html>.
- [178] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.
- [179] Zoltán Szabó and András Lőrincz. Fast parallel estimation of high dimensional information theoretical quantities with low dimensional random projection ensembles. In *Independent Component Analysis and Signal Separation (ICA)*, pages 146–153, 2009.
- [180] Barnabás Póczos and András Lőrincz. Independent subspace analysis using k-nearest neighborhood distances. *Artificial Neural Networks: Formal Models and their Applications (ICANN)*, 3697:163–168, 2005.
- [181] Jan Kybic. High-dimensional mutual information estimation for image registration. In *International Conference on Image Processing (ICIP)*, pages 1779–1782, 2004.
- [182] Gábor Takács, István Pilászy, Botyán Németh, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.
- [183] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigen-taste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

- [184] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM conference on Recommender systems (RecSys)*, pages 267–274, 2008.
- [185] Kai-Tai Fang, Samuel Kotz, and Kai Wang Ng. *Symmetric multivariate and related distributions*. Chapman and Hall, 1990.
- [186] Fabian J. Theis. Uniqueness of complex and multidimensional independent component analysis. *Signal Processing*, 84(5):951–956, 2004.
- [187] Shun-ichi Amari, Andrzej Cichocki, and Howard H. Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems (NIPS)*, 8:757–763, 1996.
- [188] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method*. Springer, 2004.
- [189] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Cross-entropy optimization for independent process analysis. In *Independent Component Analysis and Blind Signal Separation*, pages 909–916, 2006.
- [190] Zoltán Szabó and András Lőrincz. Independent subspace analysis can cope with the „curse of dimensionality”. *Acta Cybernetica (+Symposium of Intelligent Systems 2006)*, 18:213–221, 2007.
- [191] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.
- [192] Sergey Kirshner and Barnabás Póczos. ICA and ISA using Schweizer-Wolff measure of dependence. In *International Conference on Machine Learning (ICML)*, pages 464–471, 2008.
- [193] Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 313–319, 2003.

- [194] Donghui Yan, Ling Huang, and Michael I. Jordan. Fast approximate spectral clustering. In *ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 907–916, 2009.
- [195] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.
- [196] Jaime Terceiro Lomba. *Estimation of Dynamic Econometric Models with Errors in Variables*, volume 339 of *Lecture notes in economics and mathematical systems*. Berlin; New York:Springer-Verlag, 1990.
- [197] Alfredo García-Hiernaux, José Casals, and Miguel Jerez. Fast estimation methods for time series models in state-space form. *Journal of Statistical Computation and Simulation*, 79(2):121–134, 2009.
- [198] K. Rao Kadiyala and Sune Karlsson. Numerical methods for estimation and inference in Bayesian VAR-models. *Journal of Applied Econometrics*, 12:99–132, 1997.
- [199] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. Wiley-Interscience, 2008.
- [200] Zoubin Ghahramani and Geoffrey E. Hinton. Parameter estimation for linear dynamical systems. Technical report, 1996.
- [201] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82:35–45, 1960.
- [202] Arnold Neumaier and Tapio Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software*, 27(1):27–57, 2001.
- [203] Erik G. Learned-Miller and John W. Fisher III. ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, 4:1271–1295, 2003.

- [204] Francis R. Bach and Michael I. Jordan. Beyond independent components: Trees and clusters. *Journal of Machine Learning Research*, 4:1205–1233, 2003.

Short Summary in English

In my thesis I focus on (i) sparse and group-sparse coding, kernel based approximation, and (ii) independent subspace analysis (ISA) based dictionary learning.

1. I constructed a general dictionary optimization scheme for group-sparse codes; I derived novel kernel – sparsity equivalences and kernel based function approximation techniques:
 - (a) I developed a general dictionary learning technique which is (i) online, (ii) enables overlapping group structures with (iii) non-convex sparsity-inducing regularization and (iv) handles the partially observable case—previous approaches in the literature could handle two of these four desirable properties at most. I demonstrated the efficiency of the approach in 3 different applications: (i) inpainting of natural images, (ii) non-negative hierarchical matrix factorization of large scale face images, and (iii) collaborative filtering.
 - (b) I defined an extended, component-wise acting, ϵ -sparsity inducing approximation scheme in reproducing kernel Hilbert spaces (RKHS), and proved that the obtained problem is equivalent to a generalization of SVMs (support vector machine).
 - (c) I embedded SVMs to multilayer perceptrons (MLP). I proved that the well-known backpropagation method of MLPs can be generalized to the formulated multilayer SVM network.
2. I derived novel independent subspace assumption based dictionary learning problems and solution techniques:
 - (a) I coupled the active learning and the AR-IPA (autoregressive independent process analysis) tasks, and reduced the solution of the estimation problem to D-optimal ARX ('X': exogenous input) identification and ISA.
 - (b) I generalized the results of (a) to (i) the composition of linear and coordinate-wise acting nonlinear case, the so-called post nonlinear mixtures, and (ii) temporal (convolutive) mixing.
 - (c) I extended the problem of independent component analysis in case of missing observations from the former one-dimensional, i.i.d. sources to (i) multidimensional sources of (ii) not equal/-known dimensions, and (iii) relaxed the i.i.d. assumption to AR. I reduced the estimation to incompletely observed AR identification and ISA.
 - (d) I generalized the ISA problem to complex variables, and proved that under certain non-Gaussian assumption the solution can be reduced to real valued ISA.
 - (e) I extended the ISA task to the case of (i) nonparametric, asymptotically stationary source dynamics, (ii) treating the case of unknown and not necessarily equal source component dimensions. I reduced the solution of the problem to kernel regression and ISA.
 - (f) I generalized the ISA problem to convolutive mixtures, and reduced the solution of the problem to AR identification and ISA.
 - (g) Making use of the approximate distance preserving property of random projections, I presented a parallel estimation method for high dimensional information theoretical quantities. I demonstrated the efficiency of the approach in ISA.

Short Summary in Hungarian

Disszertációm a (i) ritka és csoport-ritka kódolás, kernel alapú közelítés, illetve (ii) a független alter (independent subspace analysis, ISA) feltevés és kiterjesztései melletti generátorrendszer tanulási problémával foglalkozik.

1. Általános csoport-ritka kódokhoz tartozó generátorrendszerek optimalizációjára módszert adtam; újítapusú ritkaság – kernel alapú ekvivalenciát, illetve kernel alapú függvényapproximációs módokat származtattam:
 - (a) A ritka kódokhoz tartozó generátorrendszer tanulási problémát kiterjesztettem (i) átfedő csoportstruktúrára, (ii) nem-konvex regularizációt, (iii) hiányos megfigyeléseket, és (iv) online érkező megfigyeléseket megengedő esetre – korábbi irodalombeli megközelítések ezen kívánalmak közül legfeljebb kettőt tudtak egyidejűleg kezelni. Módszerem hatékonyságát (i) természetes képek kitöltési problémáján, (ii) nagyfelbontású arcok online, hierarchikus nem-negatív mátrix faktorizációján, és (iii) kollaboratív szűrési területeken demonstráltam.
 - (b) RKHS (reproducing kernel Hilbert space)-ekben definiált ritka reprezentációs problémát kiterjesztettem az egyes koordináták mentén ható, ϵ -ritkaságokat indukáló formára. Igazoltam, hogy az így definiált alak SVM-ek (support vector machine, SVM) egy általánosított családjával ekvivalens.
 - (c) Többrétegű perceptronokba (multilayer perceptron, MLP) támasztóvektor gépeket ágyazva többrétegű SVM hálókat konstruáltam. Az összekapcsolt többrétegű perceptron hálózatra beláttam, hogy az MLP-k hibavisszaterjesztésen alapuló hangolási eljárása kiterjeszthető.
2. Független alter feltevés mellett új generátorrendszer tanulási feladatokat és megoldási technikákat származtattam:
 - (a) Az aktív tanulás és az AR-IPA (autoregressive independent process analysis) feladatot összekapcsoltam, és a megoldást D-optimalis ARX ('X': exogenous input) becslésre és ISA feladatra redukáltam.
 - (b) Az (a) munka eredményeit (i) koordinátánként ható nemlinearitásra, ún. poszt nemlineáris irányban, illetve (ii) időbeli keverést (konvolúció) megengedő irányban általánosítottam.
 - (c) A hiányosan megfigyelt független komponens keresést az eddigi 1-dimenziós, i.i.d. források esetéről kiterjesztettem (i) többdimenziós, (ii) nem feltétlenül azonos/adott dimenziós forrásokra, (iii) az i.i.d. kényszerítést egyúttal AR irányban enyhítve. A megoldást hiányosan megfigyelt AR becslésre és ISA problémára veztettem.
 - (d) Az ISA problémát általánosítottam komplex változós esetre, és a megoldást alkalmas nem-Gauss-sági feltevések esetén valós változós problémára visszavezettem.
 - (e) Az ISA feladatot kiterjesztettem (i) nemparametrikus, asszimptotikusan stacionárius forrásdinamikákra, (ii) az ismeretlen forrásdimenziók esetét is kezelve. A feladat megoldását kernel regresszióra és ISA feladatra redukáltam.
 - (f) Az ISA problémát konvolutív irányban általánosítottam, a megoldást AR becslésre és ISA feladatra redukáltam.
 - (g) A véletlen projekciók közelítő páronkénti távolságőrző tulajdonságára építve, nagy dimenziós információelméleti mennyiségek gyors, párhuzamosítható becslésére mutattam technikát és azt ISA probléma megoldására adaptáltam.